

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.043

До захисту допущено:

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«__» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»**

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Роботизована рухома охоронна платформа»

Виконав:

студент II курсу, групи ІК-91МП

Машковцев Олександр Сергійович _____

Керівник:

Доцент, к.т.н., доцент

Крилов Євген Володимирович _____

Консультант з нормоконтролю:

доцент, к.т.н., доц.,

Пасько Віктор Петрович _____

Рецензент:

к.т.н., доцент,

Катін Павло Юрійович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«__» _____ 2020 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Машковцеву Олександрю Сергійовичу

1. Тема дисертації «Роботизована рухома охоронна платформа», науковий керівник дисертації доцент, к.т.н., доцент Крилов Є.В., затверджені наказом по університету від « 26 » жовтня 2020р. № 3132-с
2. Термін подання студентом дисертації 14.12.2020
3. Об'єкт дослідження – процес проектування апаратної платформи, процес її використання.
4. Предмет дослідження – компонентна база, технології розробки та ефективність використання рухомої охоронної платформи.
5. Перелік завдань, які потрібно розробити – Проаналізувати існуючі охоронні системи, виділити їх недоліки і переваги, проаналізувати існуючі апаратні рішення необхідні для розробки платформи (системи пересування, модулі керування і обробки інформації та сенсорні системи), проаналізувати та визначити набір програмних інструментів для забезпечення роботи системи, спроектувати схему апаратного забезпечення та 3D-модель прототипу, розробити програмне забезпечення для пересування платформи, алгоритми комунікації сенсорів один з одним та із сервером.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу – 6 плакатів
7. Орієнтовний перелік публікацій – _

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перевірка на співпадіння	доцент Лісовиченко О.І.		
Нормоконтроль	доцент Пасько В.П.		

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз існуючих охоронних систем, виділення їх недоліків і переваг	01.10.2020-04.10.2020	
2	Аналіз існуючих апаратних рішень, необхідних для розробки платформи (системи пересування, модулі керування і обробки інформації та сенсорні системи)	05.10.2020-09.10.2020	
3	Аналіз та визначення набору програмних інструментів для забезпечення роботи системи	10.10.2020-11.10.2020	
4	Проектування схеми апаратного забезпечення та 3D-моделі прототипу	12.10.2020-18.10.2020	
5	Розробка програмного забезпечення для пересування платформи, алгоритмів комунікації сенсорів один з одним.	19.10.2020-08.11.2020	
8	Тестування розробленої системи	09.11.2020-13.11.2020	
9	Оформлення пояснювальної записки	14.11.2020-30.11.2020	
10	Попередній захист		
11	Нормоконтроль		
12	Перевірка на співпадіння		
13	Захист		

Студент

Олександр Машковцев

Науковий керівник

Євгеній Крилов

АНОТАЦІЯ

У цій роботі розглянуто розробку роботизованої рухомої охоронної платформи, як ефективної альтернативи класичним стаціонарним системам відеоспостереження та контролю виключних ситуацій.

Було проведено аналіз існуючих систем охорони домашнього та індустріального масштабу, виявлено їх основні недоліки і переваги. Проаналізовано актуальний перелік модулів і технологій, які доцільно використовувати в цьому проекті, було обрано найкращі альтернативи з погляду вартості і виконуваних функцій. Також було проведено огляд технологій проектування і прототипування, які значно полегшують процес розробки фінального продукту.

Результатом виконаної роботи є система, яка виконує поставлені перед нею задачі. Апаратна платформа є стійкою та сбалансованою, система збирає інформацію з сенсорів і успішно відправляє їх на сервер.

Ключові слова: патрульний робот, система охорони, роботизована платформа, Raspberry Pi.

Розмір пояснювальної записки – 83 сторінок, 63 ілюстрації, 24 таблиці, 3 додатки.

ABSTRACT

This paper considers the development process of a robotic mobile security platform as an effective alternative to the classic stationary systems of video surveillance and emergency control.

The analysis of the existing security systems of domestic and industrial scale was made. Main shortcomings and advantages of these systems were revealed. The current list of modules and technologies that should be used in this project is analyzed, the best alternatives in terms of cost and functions performed were selected. Also, there was a review of design and prototyping technologies that greatly facilitate the final product development process.

The result of the work performed is a system that performs the tasks set before it. The hardware platform is stable and balanced, the system collects information from sensors and successfully sends them to the server.

Keywords: patrol robot, security system, robotic platform, Raspberry Pi.

Explanatory note size – 83 pages, contain 63 illustrations, 24 tables and 3 applications.

**Пояснювальна записка
до магістерської дисертації**

на тему: *Роботизована рухома охоронна платформа*

Київ – 2020 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ.....	12
1.1 Об'єкт та предмет дослідження.....	12
1.2. Аналіз існуючих систем.....	12
1.3 Актуальність дослідження.....	19
1.4 Постановка задачі.....	20
Висновки до розділу.....	20
РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА РІШЕНЬ ДЛЯ РОЗРОБКИ АПАРАТНОЇ ПЛАТФОРМИ.....	22
2.1. Аналіз принципу роботи систем пересування.....	22
2.2 Розгляд основних доступних платформ пересування.....	26
2.3 Розгляд одноплатних комп'ютерів для керування РРОП.....	28
2.4 3D-друк.....	34
Висновки до розділу.....	41
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ.....	42
3.1. Розробка 3D-моделі.....	42
3.2 Архітектура системи.....	45
3.3 Програмне забезпечення компонентів рухомої платформи.....	53
3.4 Програмне забезпечення веб-платформи.....	58
4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ.....	70
4.1. Опис ідеї проекту (товару, послуги, технології) (табл. 4.1.).....	70
ВИСНОВКИ.....	80
ПЕРЕЛІК ПОСИЛАНЬ.....	82

ДОДАТКИ	83
ДОДАТОК А	84
ДОДАТОК Б.....	86
ДОДАТОК В.....	87

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IFTTT – If This Then That

РРОП – Роботизована Рухома Охоронна Платформа

RPi – Raspberry Pi

PoE – Power over Ethernet

IoT – Internet of Things

HMI – Human-Machine Interaction

STEM – Science, technology, engineering, and mathematics

ВСТУП

Патруль та нагляд протягом століть були важливою складовою охорони приватної власності. Зазвичай здійснюється пішохідне патрулювання, яке може коштувати доволі дорого і може становити небезпеку для охоронців. Окрім цього, довгі години та монотонна рутинна можуть призвести до втрати рівня безпеки, ставлячи майно та персонал під загрозу.

Рухомі патрульні охоронні платформи пропонують альтернативу для поліпшення не тільки якості безпеки, але й зменшення довгострокових витрат. Перетворення цих роботів на автономні системи збільшує їх можливості та сфери використання. Ці роботи здатні виявляти предмети / людей, які не знаходяться на місці, читати номерні знаки транспортних засобів, інтегруватися з встановленими в даний час системами безпеки тощо. Тим часом користувач може отримувати оновлення про місцезнаходження системи, статус та будь-яку підозрілу активність.

Переваги використання роботів-охоронців стають очевидними, коли працю роботів поєднують з працею людей-охоронців. Роботи здатні виконувати рутинні завдання та компенсувати будь-яку нестачу професійного персоналу. У той же час люди можуть затримувати порушників та розумно реагувати на непередбачувану злочинну діяльність. Люди та роботи, що працюють разом, дозволяють проводити більш стабільні розрахунки витрат, оскільки заробітна плата персоналу може бути зведена до мінімуму, а надійний захист в умовах, які можуть бути несприятливими для людей, посилений.

Роботизовані охоронні платформи особливо ефективні на великих ділянках, де приміщення можна розділити на місцеві охоронні зони і де мобільні роботи можуть здійснювати відеоспостереження. Кожен робот патрулює власну місцевість, забезпечуючи всебічний контроль над найвіддаленішими частинами. У той же час мобільний підрозділ співробітників служби безпеки може бути готовий швидко дістатися до місць вторгнення у відповідь на сигнал тривоги.

Поєднуючи можливості різних моделей роботів-охоронців, можна зменшити кількість персоналу на охоронюваних об'єктах, автоматизувати деякі

загальні функції безпеки, негайно посилити безпеку, коли це необхідно, та уникнути змін у якості робіт із забезпечення безпеки, що виконуються персоналом. Використовуючи охоронних робітників, компанії, що надають послуги охорони з працівниками служби безпеки, можуть швидко збільшити кількість своїх охоронюваних місць.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Об'єкт та предмет дослідження

Дослідження проводиться в області проектування апаратної частини рухомої охоронної платформи, та вибору необхідних компонентів. Дослідження у цій області допоможуть визначити переваги вибору окремих технічних рішень та доцільність використання рухомих охоронних платформ у порівнянні із стаціонарними.

Об'єктом дослідження є процес проектування апаратної платформи, а також процес її використання.

Предмет дослідження – компонентна база, технології розробки та ефективність використання рухомої охоронної платформи.

1.2. Аналіз існуючих систем



Рисунок 1.1 – Система ADT Pulse

Враховуючи дату заснування – 1874 рік, ADT, мабуть, є найбільш зрілим гравцем на ринку розумних рішень та систем охорони. Незважаючи на те, що компанія все ще будує частину свого портфолію послуг з автоматизації дому, її система Pulse є одночасно безпечною та досконалою.

Єдиним головним недоліком Pulse є ціна та трирічний контракт на обслуговування. Якщо порівнювати лише ціни, опубліковані в Інтернеті, то ADT Pulse виглядає дуже конкурентоспроможним. Але на практиці, якщо ви виберете модифіковану систему з безліччю пристроїв, які не входять до базового пакету (ADT буде закликати вас обрати саме цю систему), ціна швидко зросте.

Усі плани ADT включають цілодобовий моніторинг безпеки з допомогою безлічі резервних серверів для викликів по всій країні (що є ключем до отримання знижки від страхової компанії), консультації на місці, інсталяцію професійними техніками та цілодобове обслуговування клієнтів.

Стандартний пакет ADT пропонує лише засоби захисту від крадіжок, тоді як віддалений рівень ґрунтується на цьому, включаючи виявлення вогню, диму та CO2. Пакет із «віддаєним доступом» включає можливість включення та зняття з охорони вашої системи безпеки через Інтернет, отримання електронної пошти та текстових сповіщень, а також дозволяє використання резервної копії CellGuard, яка забезпечує передачу даних та зв'язок на основі стільникового зв'язку у випадку, якщо ваш основний Інтернет-канал виходить з ладу.



Рисунок 1.2 – Мобільний додаток ADT Pulse

Пакет «Control» бере все з попередніх рівнів і додає основи домашньої автоматизації, включаючи клімат-контроль, освітлення та управління приладами, модернізовану сенсорну клавіатуру та дистанційні (розумні) дверні замки. Нарешті, рівень відео додає відеозахист у режимі реального часу та інші

додаткові додаткові можливості, такі як додаткові клавіатури, якщо це необхідно.

ADT підтримує цілий ряд камер безпеки від сторонніх виробників, включаючи внутрішні камери, якщо ви цього захочете. Тому ви можете поговорити зі своїм техніком про різні варіанти, якщо базовий вам не підходить.

Одною з основних переваг є мобільний додаток Pulse. Він підтримує пристрої Android та iOS, і якщо веб-портал використовується для повного управління системою Pulse, мобільний додаток – це ваша система контролю активності. Ви можете отримати доступ до мобільного додатку, використовуючи свій основний логін веб-порталу ADT або чотиризначний PIN-код.



Рисунок 1.3 – Система FrontPoint Safe Home

У попередній версії системи FrontPoint в якості основного концентратора використовувалася панель із сенсорним екраном, але остання версія використовує елегантну комбінацію концентратора та клавіатури, хоча сенсорну панель можна придбати за 95,99 доларів. У верхній частині є світлодіодний індикатор, який горить зеленим, коли система знята з охорони, червоним – коли вона перебуває в режимі відключення, жовтим – коли вона знаходиться в режимі охорони, і синім – коли є проблема. На задній панелі розміщений порт LAN для дротового підключення, порт живлення та кнопка «reset», а на базі є динамік для голосових підказок, що повідомляє про стан системи.

На борту система має стільниковий зв'язок, Wi-Fi та Z-Wave, дуже гучну сирену та резервну батарею, здатну підтримувати роботу системи протягом 24 годин. Окрім компонентів домашньої безпеки, концентратор може керувати численними пристроями домашньої автоматизації, включаючи замки, термостати, ліхтарі та контролери гаражних воріт.

Усі датчики, що входять до комплекту Safe Home Elite, попередньо підключені до концентратора та використовують двосторонній скотч для зручності монтажу. Датчики дверей / вікон порівняно тонкі (1,8 на 0,7 на 0,3 дюйма) і живляться від батареї CR1632, що розрахована на 6 років. Датчик руху працює від однієї батареї CR123A і становить 4,5 на 2,5 на 2,0 дюйма. Діапазон – 44 фути з 90-градусним полем зору. Датчик розбиття скла має розміри 3,1 на 4,2 на 1,7 дюйма, працює від двох батарей CR123A і може розпізнавати звук розбитого скла на відстані до 20 футів. Датчик диму і тепла круглий і має розміри 5,6 на 2,3 дюйма. Він живиться від двох батарейок 3В, які герметичні та не замінюються, але розраховані на 10 років роботи.

Frontpoint пропонує дві внутрішні камери (стандартну та преміальну), зовнішню камеру та відеокамеру у дзвінок. Кімнатна камера у формі шайби має роздільну здатність 1080p, поле зору 113 градусів, Wi-Fi 2,4 ГГц та 5 ГГц, двосторонній звук та виявлення руху. Внутрішня камера преміум-класу має роздільну здатність 1080p та 180-градусний огляд і функціонує як динамік Bluetooth. Зовнішня камера – це захищена від непогоди камера IP65, яка фіксує відео з роздільною здатністю 1080p та має поле зору 117 градусів та Wi-Fi 2,4 ГГц та 5 ГГц. Дверний дзвінок має роздільну здатність 720p та поле зору 180 градусів, містить Wi-Fi радіо 2,4 ГГц, динамік та мікрофон для двостороннього звуку та датчик руху.

Мобільний додаток FrontPoint (для Android та iOS) та веб-портал працюють на Alarm.com. Вони обидва дозволяють вам поставити на озброєння та зняти з озброєння охоронну систему, переглянути відео з камери та дзвінка (у прямому ефірі та на записі), активувати сповіщення та налаштовувати нагадування про встановлений будильник. Мобільний додаток відкриває головний екран, на якому відображається назва вашого будинку та вибір

контролерів, які одним натисканням кнопки переведуть систему в озброєний або роззброєний стан. Ви також можете запускати різні сценарії за допомогою голосових команд Alexa або Google Assistant, але система не підтримує аплети IFTTT.

На головному екрані також є вкладки для всіх встановлених камер та дверних дзвінків. Можна додавати користувачів та керувати сповіщеннями у мобільному додатку, але веб-портал пропонує додаткові налаштування, які дозволяють вам планувати отримання сповіщень для певних сенсорів, налаштовувати автоматизацію та переглядати історію всіх дій.



Рисунок 1.4 – Система SimpliSafe

Одна з найліпших відмінностей SimpliSafe полягає в тому, що це повністю налаштовувана система з п'ятьма доступними ціновими пакетами. Найбільшим є Haven (\$ 489), який постачається з 14 апаратними компонентами, включаючи базову станцію, бездротову клавіатуру, пульт дистанційного керування, два датчики руху, чотири датчики входу дверей / вікон, кнопку паніки, 105-децибел сирену, датчики диму та чадного газу, датчики води та замерзання, а також ярлики та наклейки на вікна SimpliSafe. Найменш дорогий пакет Foundation (229 доларів США), який призначений для невеликих будинків і орієнтований виключно на базову домашню безпеку, включає базову станцію, бездротову клавіатуру, один датчик руху, один датчик входу, а також ярлик та наклейки на подвір'я.

Якщо ви хочете додати одну або кілька камер безпеки SimpliCam (система підтримує до чотирьох), вартість складає 99 доларів за кожную. Video Doorbell Pro – це вбудований розумний дзвінок 1080p із камерою, який можна додати за 169 доларів, тоді як Smart Lock за 99 доларів відкриває можливості безключового доступу.

Додаткові компоненти також можна придбати окремо, щоб створити власний пакет, або розширити існуючий. Наприклад, якщо у вас багато вікон на першому поверсі, вам може знадобитися датчик розбиття скла (34,99 доларів США) або додаткові датчики входу (14,99 доларів США кожен; система підтримує до 100 датчиків). Або якщо у вашому домі є кілька входів, ви можете обрати додаткові клавіатури (69,99 доларів США).

Якщо вам потрібне універсальне рішення моніторингу будинку, яке охоплює не лише охорону, додавання сенсорів навколишнього середовища має сенс. Детектори диму коштують 29,99 доларів США кожен, і ви можете підключити їх до свого плану моніторингу, який сповіщатиме вас електронною поштою або текстовим повідомленням про спрацьовування сигналізації та надсилатиме пожежників разі пожежі. Датчики заморозки (\$ 29,99 кожен) попереджають вас, коли температура опускається нижче певної точки, щоб запобігти прориву труб, а датчики води (\$ 19,99 кожен) виявляють і попереджають про витoki та повені.



Рисунок 1.5 – Система Wyze Sense Starter Kit

Початковий набір Sense постачається з двома невеликими контактними датчиками, датчиком руху та центральним пристроєм. Контактні датчики мають розміри 1,1 на 1,2 на 0,9 дюйми (HWD). Компоненти оснащені двостороннім скотчем для легкої установки, а на лицьовій стороні датчика є крихітний світлодіодний індикатор, який блимає червоним кольором, коли він спрацьовував. Датчик живиться від батареї кнопки CR1632, яка розрахована на тривалість роботи до року, перш ніж потребуватиме заміни. Додаткові контактні датчики можна придбати за 19,99 доларів за два пакети.

Датчик руху має розміри 1,2 на 1,2 на 0,9 дюйми (HWD). Він також має крихітний червоний світлодіодний індикатор стану, але живиться від більшої батареї CR2450, яка повинна тривати до одного року. На торці є чорний круговий датчик, який забезпечує виявлення руху на висоті до 26 футів. Обидва датчики мають невеликий отвір для сбросу конфігурації збоку. Додаткові датчики руху коштують \$ 5,99 кожен.

Міст підключається до заднього USB-порту камери Wyze Cam або Wyze Cam Pan (не входить в комплект). Він має розміри 1,2 на 1,2 на 0,7 дюйма (HWD) і має роз'єм USB типу A з одного боку та висувну вкладку з іншого, яка використовується для його виймання з камери. Його світлодіодний індикатор блимає синім, коли міст перебуває в режимі сполучення, постійно синім, коли він працює належним чином, і блимає жовто-блакитним під час спроби підключення.

Коли ви встановлюєте міст на камеру, ви можете змусити датчики запускати камеру для початку запису та безкоштовно зберігати відео в хмарі протягом чотирьох днів. Або ви можете встановити карту microSD у камеру для локального зберігання відео. Датчики також можуть бути використані для запуску сторонніх інтелектуальних пристроїв, таких як камери та ліхтарі, за допомогою аплетів IFTTT та Alexa Routines. Однак Apple Home Kit не підтримується.

Датчики використовують той самий мобільний додаток Wyze (для Android та iOS), що і камери Wyze Cam та Wyze Cam Pan. Він відкриває головний екран, який містить вкладки для всіх встановлених пристроїв Wyze з їхніми іменами та

статусом (Інтернет, Офлайн). Достатньо доторкнутися до будь-якого датчика, щоб переглянути список дій із деталями, наприклад, коли датчик був відкритий і як довго, або коли було виявлено рух, а також ескіз відеокліпу, якщо ви налаштували параметри відео датчика.

1.3 Актуальність дослідження

Автономні охоронні роботи поєднують у собі безліч різних технологій для виконання операцій з охорони та спостереження, таких як звітування, моніторинг, розслідування та виявлення зловмисників. Сьогодні безпека є основною проблемою для багатьох підприємств. Захист активів, майна, місцезнаходження персоналу – це повсякденна робота майже всіх фахівців з безпеки, що працюють у різних галузях. У багатьох професіоналів охорони багато робочого часу витрачається на моніторинг та патрулювання, при цьому вичерпується час на контроль правопорушень чи підозрілої діяльності. Це може бути особливо складним завданням при роботі з великими територіями, які потребують кількох охоронців. Таким чином, організації з різних секторів зараз починають розглядати автономні роботи з безпеки як спосіб розширення або посилення своїх охоронних систем. Нижче наведено кілька причин, чому автономні роботи безпеки користуються великим попитом.

- Мобільність: Автономні роботи мають можливість стабільно рухатися разом в складних середовищах, виявляти та повідомляти про підозрілу активність в цих середовищах. Вбудовані навігаційні системи дозволяють роботам правильно орієнтуватись у географічному розташуванні, а комбінація датчиків і камер допомагає в навігації в складній місцевості.
- Додаткова робоча сила: Автономні роботи безпеки можуть використовуватися як додаткова робоча сила для охоронців. Ці роботи дозволяють виконувати більше завдань, застосовуючи однакову кількість зусиль, без потреби у збільшенні кількості охоронців.
- Економічна ефективність: автоматизовані патрульні роботи є дуже хорошим варіантом для підприємств та організацій, які потребують більшої системи безпеки, але не можуть знайти бюджет для підтримки та

найму нового працівника. Ці роботи в поєднанні з іншими перевагами та технологіями IoT, такими як передбачення даних та датчиками, можуть працювати навіть значно дешевше, ніж вартість додаткового персоналу.

- Більш тривала робота: На відміну від охоронців-людей, автономні охоронні роботи не страждають від стресу та втоми і можуть працювати довший час. Це робить їх ідеальними для підтримки безпеки або для резервного копіювання у випадках людської помилки.

1.4 Постановка задачі

Метою магістерської дисертації є вирішення задачі проектування моделі автономної рухомої охоронної платформи, покращення цінової політики завдяки правильно підібраній компонентній базі, доведення можливості використання такої системи як в домашній середі, так і в індустріальній.

Для досягнення поставленої мети в роботі поставлені такі задачі:

- Проаналізувати існуючі охоронні системи, виділити їх недоліки і переваги
- Проаналізувати існуючі апаратні рішення необхідні для розробки платформи (системи пересування, модулі керування і обробки інформації та сенсорні системи)
- Проаналізувати та визначити набір програмних інструментів для забезпечення роботи системи
- Спроекувати схему апаратного забезпечення та 3D-модель прототипу
- Розробити програмне забезпечення для пересування платформи, алгоритми комунікації сенсорів один з одним та із сервером

Висновки до розділу

В даному розділі було проведено аналіз існуючих на ринку систем, було виявлено їх переваги та недоліки. Серед основних переваг таких систем можна виділити можливість під'єднання додаткових сенсорів і систем контролю, комфортні у користуванні мобільні додатки та веб-платформи, велику кількість можливих кастомізацій у рамках нативних рішень для обраних платформ,

можливість роботи систем в умовах відсутності основного живлення, використовуючи резервні батареї, легкий процес налаштування і використання, інтеграції із найпопулярнішими голосовими асистентами. Серед недоліків можна виділити доволі затратні системи підписок та загальну велику вартість систем, відсутність мобільності і необхідність встановлення датчиків на кожному контролюємому об'єкті.

Виходячи з цих даних, було сформовано, мету та основні задачі, необхідні для реалізації проекту.

РОЗДІЛ 2. ВИБІР ТЕХНОЛОГІЙ ТА РІШЕНЬ ДЛЯ РОЗРОБКИ АПАРАТНОЇ ПЛАТФОРМИ

2.1. Аналіз принципу роботи систем пересування

Мобільному роботу необхідні певні механізми пересування, щоб він міг рухатися по своєму оточенню. Існує декілька механізмів для досягнення цієї мети. Механізми руху ніг роботів часто надихаються біологічними системами, які є дуже успішними у пересуванні по різних типах поверхностей.

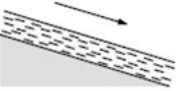
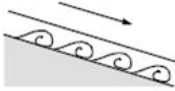

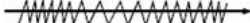








Type of motion	Resistance to motion	Basic kinematics of motion
Flow in a Channel 	Hydrodynamic forces	Eddies 
Crawl 	Friction forces	Longitudinal vibration 
Sliding 	Friction forces	Transverse vibration 
Running 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Jumping 	Loss of kinetic energy	Oscillatory movement of a multi-link pendulum 
Walking 	Gravitational forces	Rolling of a polygon 

Рисунок 2.1 – Природні механізми пересування

На рис. 2.1 наведено деякі механізми руху, які зустрічаються в природі. Звичайно, їх набагато більше, наприклад шестинога ходьба паличної комахи, що часто є парадигмою шестиногих роботів. Але скопіювати ці механізми дуже важко за декількох причин. Основними проблемами є механічна складність ніг, стійкість і кількість споживаної електроенергії. З іншого боку, існує колісна локомоція. Колеса – це винахід людини і дуже популярний засіб пересування транспортних засобів. З першого погляду, в природі немає подібних систем, але ходіння людини може апроксимуватися як кочення (рис. 2.2).

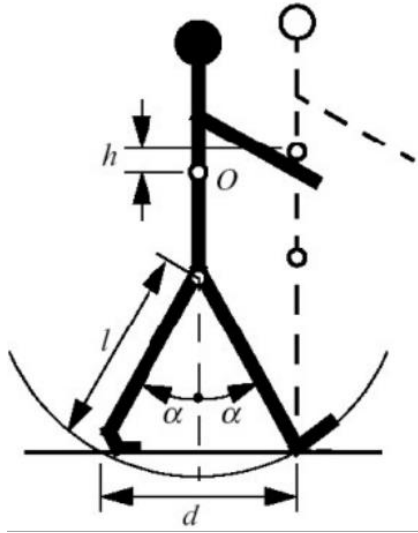


Рисунок 2.2 – Механізм ходіння людини

Для кожного виду пересування, без різниці, використовуватиметься колесо, ноги чи інша концепція – є три основні проблеми: стабільність, характеристики контакту з землею та тип середовища пересування. Основними атрибутами стабільності є кількість і геометрія точок контакту, сила тяжіння і нахил місцевості. Характеристики контакту із землею залежать від типу точки контакту (у випадку з ногами – це відбиток стопи), кут контакту з землею і тертя між роботом і поверхнею. Атрибутами типу середовища є структура середовища (наприклад, у випадку жорсткого середовища), і саме середовище (наприклад, вода, повітря, тверда або м'яка земля).

У нашому випадку платформа буде використовуватися для роботи у приміщенні, на жорсткій та рівній поверхні. Виключення становлять різні покриття, такі як килими, лінолеум, стики між приміщеннями. Через це будемо використовувати колісну або гусеничну платформу пересування.

Пересувні колісні та гусеничні роботи мають мінімум два двигуни, які використовуються для приведення в рух і керування роботом. Ентузіасти зазвичай обирають ковзаюче рульове через простоту інсталяції, конфігурації та управління. Використання третього заднього колеса, як правило, запобігає падінню робота. Чотириколісні роботи мають два, або чотири двигуна і використовують рульове управління. Шестиколісні роботи найчастіше мають два, чотири або шість двигунів. Збільшення кількості приводних двигунів

допомагає роботу підніматися на більш круті нахили, за допомогою підвищеного крутного моменту.

Додавання коліс без двигуна часто призводить до втрати ваги з ведучих коліс, що призводить до ковзання та втрати тяги. На рис. 2.3 центральне колесо, обране помилково як приводне колесо, часто втрачає контакт із землею.

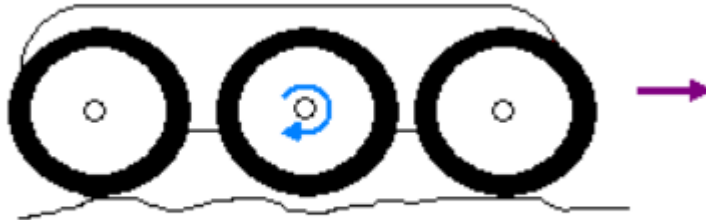


Рисунок 2.3 – Колісна база із центральним ведучим колесом

Система колій (або зв'язування всіх коліс із шестернями або приводними ремнями) також допомагає запобігти цьому. Колії – не обов'язково кращий вибір, ніж використання кількох приводних коліс. Виробники танків у всьому світі виробляють як колісні, так і гусеничні моделі танків, і обидві концепції демонструють майже однакові показники. Більшість користувачів погоджуються, що танкові колії – далеко не самий вишуканий спосіб транспортування і мають тенденцію "рвати" землю під собою. Роблячи щільний поворот або повертаючись на місці, протектори танка стикаються з значно більшим опором, ніж колеса, оскільки обидві половини притискаються до землі перпендикулярно радіусу повороту. Саме через це такі платформи споживають дуже багато електроенергії.

Можливо розробити самобалансуючого робота, використовуючи два колеса. Або можливо розробити більш стабільну платформу, використовуючи чотири або більше.

Два ведучих колеса використовуються для приведення в рух і повороту робота (рульове керування), а одне або два холості колеса запобігають падінню робота вперед або назад. Колесо «холостого ходу» може бути роликом або кулею.



Рисунок 2.4 – Платформи на двох ведучих колесах та одному колесі
«холостого ходу»

Омні-платформи (або Omniwheels) зазвичай мають три, розташовані трикутно, або чотири, розташовані під 90 градусів колеса. Omniwheels мають невеликі роликові колеса, які вільно обертаються. Зазвичай ведучим є центральне колесо. Це дозволяє рухати колесо вперед, але ковзати вбік.

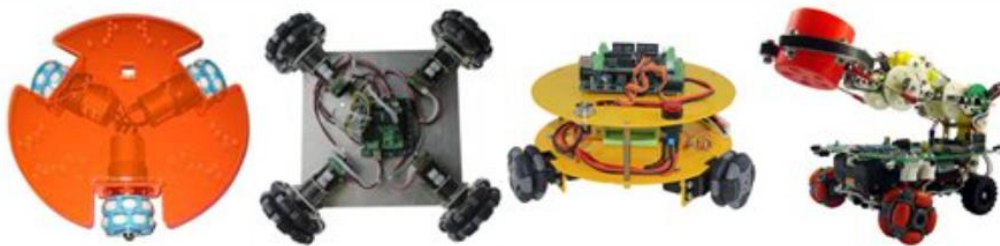


Рисунок 2.5 – Платформи Omniwheels

Правильний розмір колеса дуже важливий, і слід подбати, щоб вибрати правильний для кожного конкретного робота. Потрібно врахувати два основних рівняння:

- швидкість = кутова швидкість (колеса) x радіус (колеса)

Це означає, що і радіус колеса, і кутова швидкість, з якою воно обертається, впливатимуть на швидкість руху вперед.

- сила (що діє колесом на поверхню) = крутний момент (двигуна) / радіус (колеса)

Щоб колісний робот рухався або піднімався по нахилу, колесо прикладає горизонтальну силу на поверхню. Якщо вам потрібно прикласти велику силу (оскільки ваш робот важкий), то вам потрібно збільшити крутний момент або зменшити радіус колеса. Збільшення крутного моменту, як правило, дороге, а збільшення радіуса зазвичай означає важче колесо, що вимагає більше крутного

моменту, але також зменшує максимальну швидкість (оскільки воно повинне обертатися швидше). Дуже малий діаметр коліс також може бути недоцільним.

2.2 Розгляд основних доступних платформ пересування



Рисунок 2.6 – Платформа Keyestudio

Комплект Keyestudio (коштує близько 20 доларів) включає чотири двигуни, а також обладнання для кріплення на шасі, що робить його чудовим комплектом для тих, хто тільки починає свій проект. Гумові шини сприяють тязі, а пластикові хаби призначені для щільного ковзання на моторизовані осі. Серед основних переваг – наявність проводів для підключення у комплекті, гнучкий діапазон напруги. Серед недоліків – доволі сильно хитається при пересуванні.



Рисунок 2.7 – Гусенична платформа «Tamiya Tracked Vehicle»

Гусенична платформа Tamiya (коштує близько 25 доларів) має коробку передач з моноприводом, а це означає, що вона може їздити лише по прямій лінії, але не надто важко перейти до установки з декількома трансмісіями. У комплекті є майже все, що потрібно для її складання, навіть вимикач живлення.



Рисунок 2.8 – Платформа Wild Thumper

Wild Thumper – це найкраще 6-колісне шасі для роботів у своєму ціновому діапазоні.

Кожен із шести двигунів встановлений на незалежній підвісці та оснащений сталевими коробками передач із відношенням 34:1. Двигуни з'єднані між собою на клемній колодці всередині анодованої металевої рами товщиною 2 мм і ними можна керувати за двома каналами (лівим і правим). Весь корпус перфорований 10-міліметровими отворами, кріпильними отворами діаметром 4 мм, а в шасі є достатньо місця для батарей, драйверів та іншого допоміжного обладнання. Шасі поставляється з шістьма вузькими 120-мм позашляховими колесами.

Оскільки двигунів усього шість, кожен канал драйвера повинен мати змогу подавати велику кількість струму для приведення моторів у дію. Це платформа диференціального приводу, що означає, що поворот здійснюється простим рухом однієї сторони швидше або повільніше, ніж інший, або поворотом обох сторін у протилежні сторони. Оскільки "точка повороту" по суті складається з протилежних трьох шин, для поступового повороту буде потрібно менше потужності, ніж для різкого, і чим більше енергії вийде передати, тим більш маневреною буде платформа.

2.3 Розгляд одноплатних комп'ютерів для керування РРОП



Рисунок 2.9 – Raspberry Pi 4 Model B

Raspberry Pi 4 Model B оснащена найновішим високопродуктивним чотирьохядерним 64-розрядним процесором Broadcom 2711, Cortex A72 з тактовою частотою 1,5 ГГц. Плата розроблена на 20% більш енергоефективною та забезпечує на 90% більшу продуктивність, ніж її стара версія. Оновлення апаратного забезпечення на RPi4 розроблено для швидшої продуктивності не тільки за рахунок нових варіантів SDRAM 1 ГБ / 2 ГБ та 4 ГБ SDRAM, але також за рахунок дводіапазонної 2,4 ГГц та 5 ГГц бездротової локальної мережі 802,11 b / g / n / ac та можливості PoE через окремий PoE HAT. На додаток до USB 3.0, вони покращили швидкість передачі даних в 10 разів. Можливі версії з 1, 2 і 4 ГБ оперативної пам'яті.

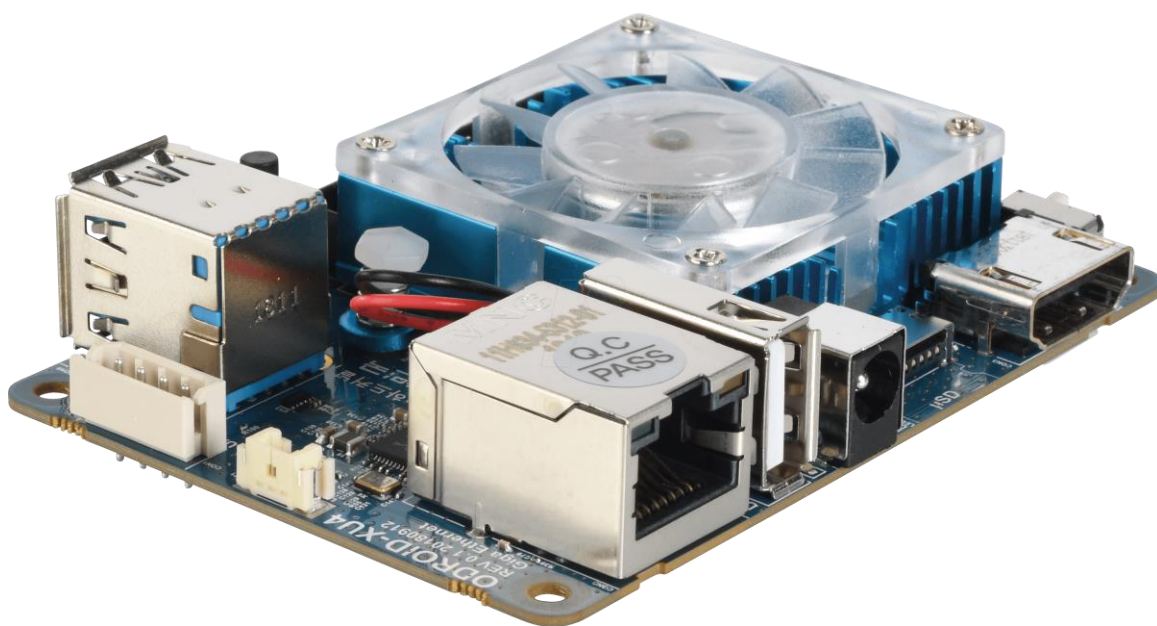


Рисунок 2.10 – Odroid-XU4

Odroid-XU4 – це економічно вигідний варіант з високою потужністю, з одним гігабітним портом Ethernet для швидкого підключення, HDMI 1.4a для дисплея, а плата навіть постачається з активним кулером та адаптером живлення. Він оснащений потужним восьмиядерним процесором Samsung Exynos 5422 Cortex-A15 з тактовою частотою 2 ГГц та графічним процесором Mali-T628 MP6 з підтримкою OpenGL ES 3.1 та OpenCL 1.2. ODROID-XU4 також має 2 ГБ пам'яті, два порти USB 3.0, але немає Bluetooth для бездротового підключення. На платі можна запустити останню версію Ubuntu, Android 4.4 KitKat, 5.0 Lollipop та 7.1 Nougat. ODROID-XU4 працює на основі технології ARM big.LITTLE та рішення гетерогенної багатопроцесорної обробки (HMP).

Впроваджуючи інтерфейси eMMC 5.0, USB 3.0 та Gigabit Ethernet, ODROID-XU4 може похвалитися вражаючою швидкістю передачі даних – функцією, яка все частіше потрібна для підтримки розширеної потужності обробки на ARM-пристроях. Для завантаження ОС потрібна карта MicroSD або модуль eMMC.

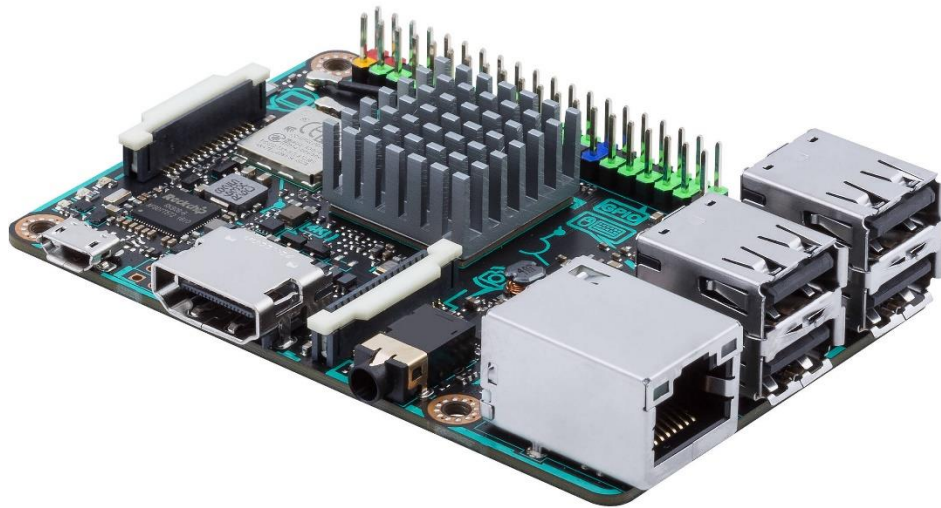


Рисунок 2.11 – ASUS Tinker Board RK3288

Так само, як ODROID-XU4, ASUS Tinker Board можна використовувати як щоденний комп'ютер з більш ніж достатньою потужністю для базового редагування зображень, потокового передавання відео у форматі Full HD, перегляду веб-сторінок, прослуховування музики та навіть деяких ігор. На платі встановлений Rockchip RK3288, який є сучасним чотирьохядерним процесором на базі ARM, який можна знайти в багатьох планшетах та мультимедійних програвачах. Завдяки 2 ГБ пам'яті та графічному процесору Mali-T764, плата може відтворювати HD та UHD відео зі швидкістю 30 кадрів в секунду за допомогою включеного медіаплеєра з підтримкою апаратного прискорення. Плата також включає 40-контактний інтерфейс GPIO, гігабітне підключення до локальної мережі, підключення DSI MIPI для дисплеїв та сенсорних екранів, а також підключення CSI MIPI для підключення до сумісних камер, що робить його чудовим для Інтернету речей.

Якщо потрібний недорогий одноплатний комп'ютер для налаштування мережевого сховища, файлового сервера чи чогось іншого, одноплатний

комп'ютер Libre Renegade буде гарним вибором. Він має чотирибітний 64-розрядний процесор ARM Cortex-A53 1,4 ГГц, графічний процесор 4K Ultra HD ARM Mali-450 та 2 Гб оперативної пам'яті DDR4. Ця система насправді на 40% швидше, ніж Raspberry Pi. Комп'ютер Libre's Renegade навіть оснащений швидким гігабітним Ethernet, якого не зустрінеш на багатьох платах.

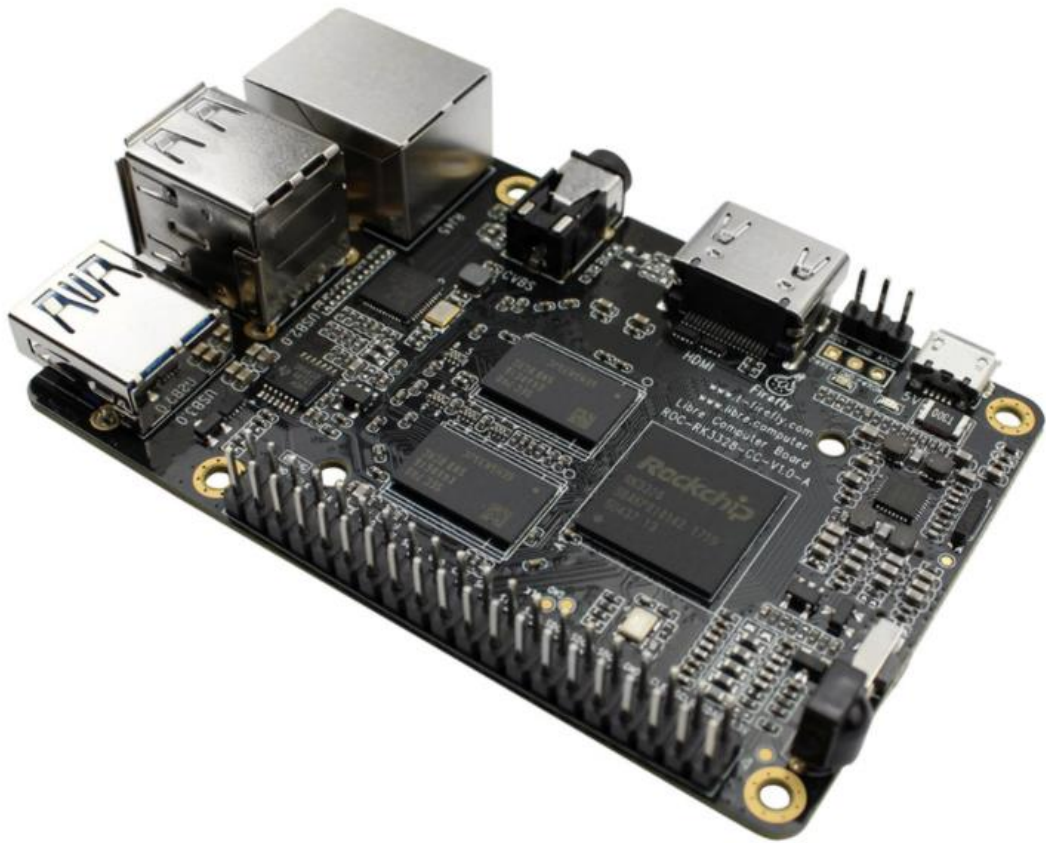


Рисунок 2.12 – Libre Renegade

Плата Raspberry Pi Zero W розширює функціональність плати Pi Zero завдяки бездротовій локальній мережі та Bluetooth. Багато в чому це ідеальний одноплатний комп'ютер для епохи Інтернету речей. Беручи до уваги його крихітний розмір, ви можете бути здивовані, дізнавшись, що він поставляється з процесором Broadcom BCM2835 з тактовою частотою 1 ГГц та 512 МБ LPDDR2 SDRAM. За надзвичайно низьку вартість ви можете отримати неймовірно універсальний Wi-Fi одноплатний комп'ютер, який стоїть на плечах найактивнішого та найкориснішого співтовариства існуючих ентузіастів одноплатних комп'ютерів. Розміри Raspberry Pi Zero W мають розмір всього 6,5 см × 3 см × 0,5 см, і він споживає настільки мало енергії, що навіть помірно великий акумулятор може тримати його в роботі дуже довго.



Рисунок 2.13 – Raspberry Pi Zero W

Banana-Pi – це комп'ютер розміром з кредитну картку з чудовими функціями. Він вироблений китайською компанією Shenzhen co, Ltd. Raspberry Pi вплинув на дизайн комп'ютерного обладнання banana-Pi. Banana-Pi може працювати в багатьох операційних системах, таких як Linux, Android, Debian та Ubuntu. Banana-Pi BPI-M4 використовує чіп-структуру Realtek RTD1395, це 64-бітний зменшений одноплатний ПК A53. Він має 1 ГБ / 2 ГБ оперативної пам'яті та 8 ГБ eMMC. BPI-M4 має 4 порти USB 2.0, 1 порт USB TYPE C, 1 порт HDMI, звуковий роз'єм f1. Є підтримка M.2 Key E інтерфейсу PCI-E 2.0.

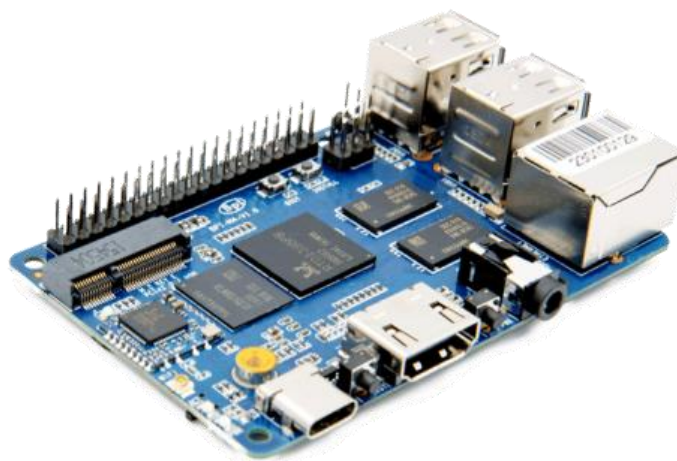


Рисунок 2.14 – Banana Pi BPI-M4

Arduino – це програмована плата з відкритим кодом, яка може бути інтегрована в найрізноманітніші проекти. Ця плата містить мікроконтролер, який можна запрограмувати для управління різними пристроями та сприйняття великої кількості інформації. Arduino здатний взаємодіяти з великим набором вихідних пристроїв, таких як світлодіоди, двигуни та дисплеї. Завдяки своїй гнучкості та низькій вартості, Arduino став дуже популярним вибором для інженерів для створення інтерактивних апаратних проектів.

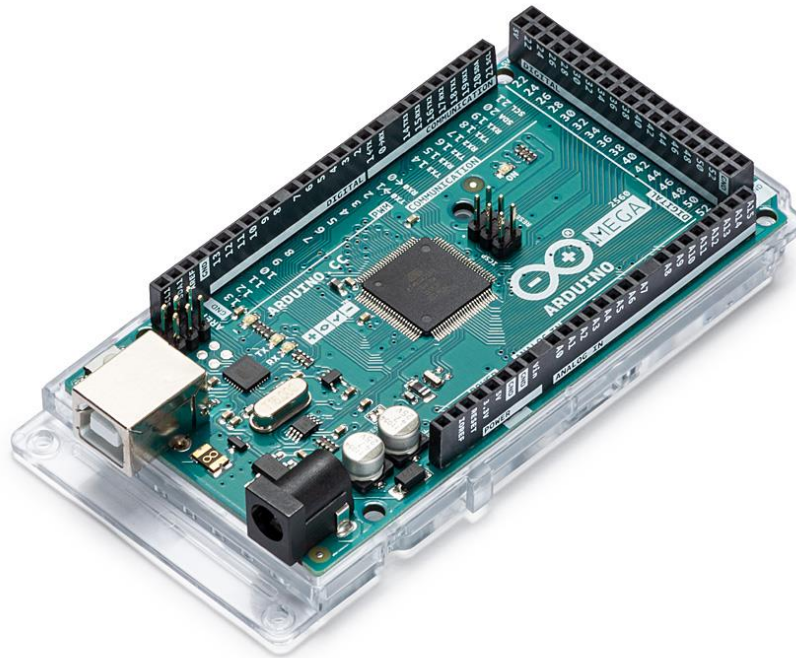


Рисунок 2.15 – Arduino Mega 2560

Arduino був представлений в 2005 році в Італії Массімо Банзі як спосіб для «не інженерів» отримати доступ до недорогого, простого інструменту для створення апаратних проектів. Оскільки це open-source проект, він випускається під ліцензією Creative Commons, яка дозволяє кожному створити власну плату.

Мова програмування Arduino відрізняється від інших мов програмування, оскільки її легко вивчити та вона приховує всі труднощі написання програм у мікроконтролерах, надаючи пакет, який спрощує та обгортає всі важливі деталі. Деякі ключові особливості та переваги Arduino:

- Зручне для користувача середовище
- Економічно вигідний
- Відкритий і розширюваний код

- Крос-платформа

2.4 3D-друк

3D-друк – це процес створення фізичного об’єкта з цифрової моделі. 3D-друк – це адитивний процес. Шари пластику нарощуються один за одним, щоб створити предмет. Фактично, котушки із пластикових ниток або лотків із смолою перетворюються у фізичні предмети.



Рисунок 2.16 – 3D-принтер Ender 3 Pro

3D-друк почали використовувати студенти, підприємці, любителі та масові фабрики, оскільки він дозволяє перетворити цифровий дизайн на матеріальний об’єкт. Це знайшло широкий спектр використання.

Лікарі можуть надрукувати фізичні моделі частин тіла пацієнта для кращої візуалізації процедур та демонстрації практики, інженери на фабриках можуть

створювати спеціальні пристосування, які заощаджують час та зменшують травми під час виробничого процесу. Громади сприяють створенню просторів, які навчають навичкам STEM та допомагають у відкритті нового бізнесу, створюючи тим самим нові робочі місця та можливості. НАСА має 3D-принтер на Міжнародній космічній станції, і астронавти можуть створювати власні інструменти, не маючи необхідності відправляти їх у космос.

Хоча в основному використовується для створення пластикових предметів, 3D-друк може також генерувати металеві предмети, хоча це набагато дорожчий і набагато рідше процес, ніж пластиковий 3D-друк.

Аналогія з друком не зовсім вірна. Комп'ютерні принтери, як правило, опрацьовують по одному рядку за цикл. 3D-принтери працюють більше як плотери, рухаючи друкуючу головку вздовж осі X і Y, щоб намалювати візерунок. У випадку з 3D-принтером малюнок зазвичай малюють пластиком, а не чорнилом. Що робить вироби, зроблені на 3D-принтері тривимірними, так це те, що як тільки малюнок намальовано, друкуюча головка рухається вгору (або поверхня друку рухається вниз), а інший малюнок малюється поверх першого.

Існує кілька типів 3D-принтерів, але ми зупинимось на двох: моделювання з плавленням осадженням (або FDM) та стереолітографія (або SLA).

FDM використовує рулони ниток як вихідний матеріал. Як правило, застосовуються нитки товщиною 1,75 або 2,85 міліметрів, згорнуті на шпindel. Принтер FDM нагріває нитку, видавлює її через екструдерне сопло і кладе шари на поверхню для друку. Шари неймовірно тонкі, і оскільки кожен розплавлений шар укладається поверх попереднього шару, він частково наплавлюється.

Через певний період часу (іноді доволі великий) модель предмета, яка складається з безлічі шарів, з'являється на столі.

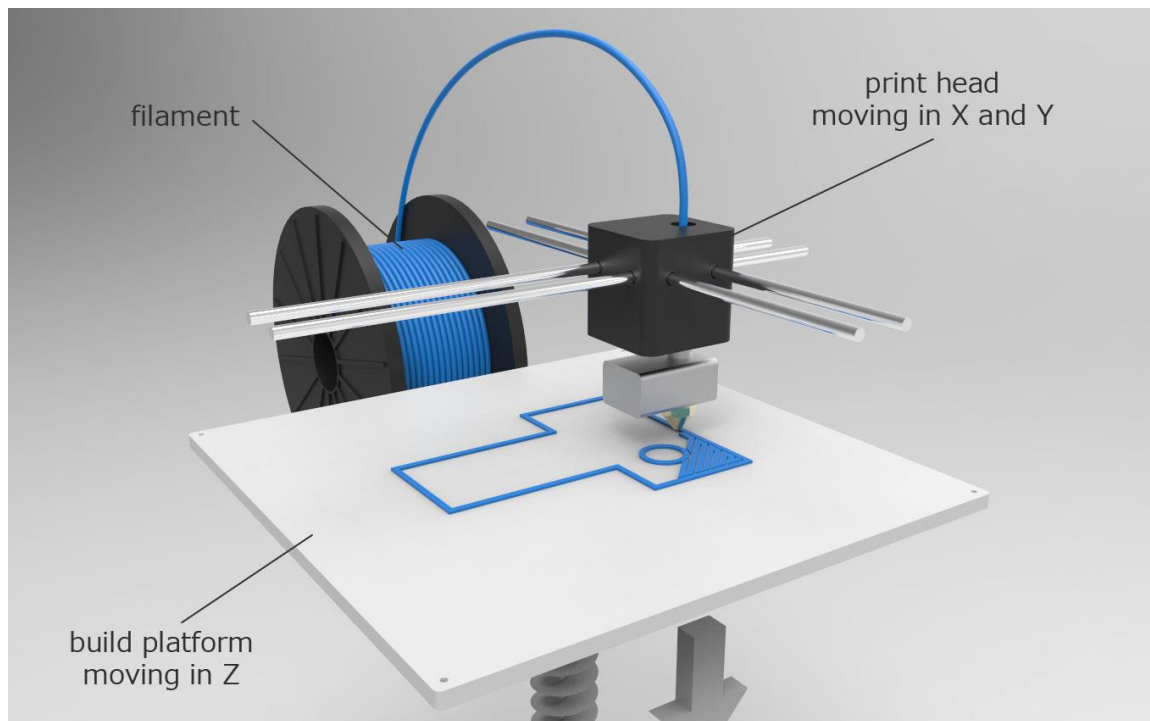


Рисунок 2.17 – FDM друк

SLA використовує рідку смолу як вихідний матеріал. Лоток для нарощування опускається в смолу (зазвичай догори дном), а світло (іноді від РК-дисплея, іноді від ультрафіолетового лазера) створює в смолі хімічну реакцію, яка спричинює її твердіння. Оскільки кожен шар піддається впливу світла, принтер трохи піднімає будівельну платформу із смоляного пулу, піддаючи світло наступному шару.

FDM є найпоширенішою формою екструзії матеріалів 3D-друком. SLA є найпоширенішою формою легкої полімеризації 3D-друку. Ці два способи друку досягли рівня витрат на досить низькому рівні, щоб їх могли собі дозволити звичайні споживачі, любителі, освітяни, підприємці та малий бізнес, але вони, як правило, обмежуються виробництвом пластмас, пластикових композитів та нейлоноподібних матеріалів.

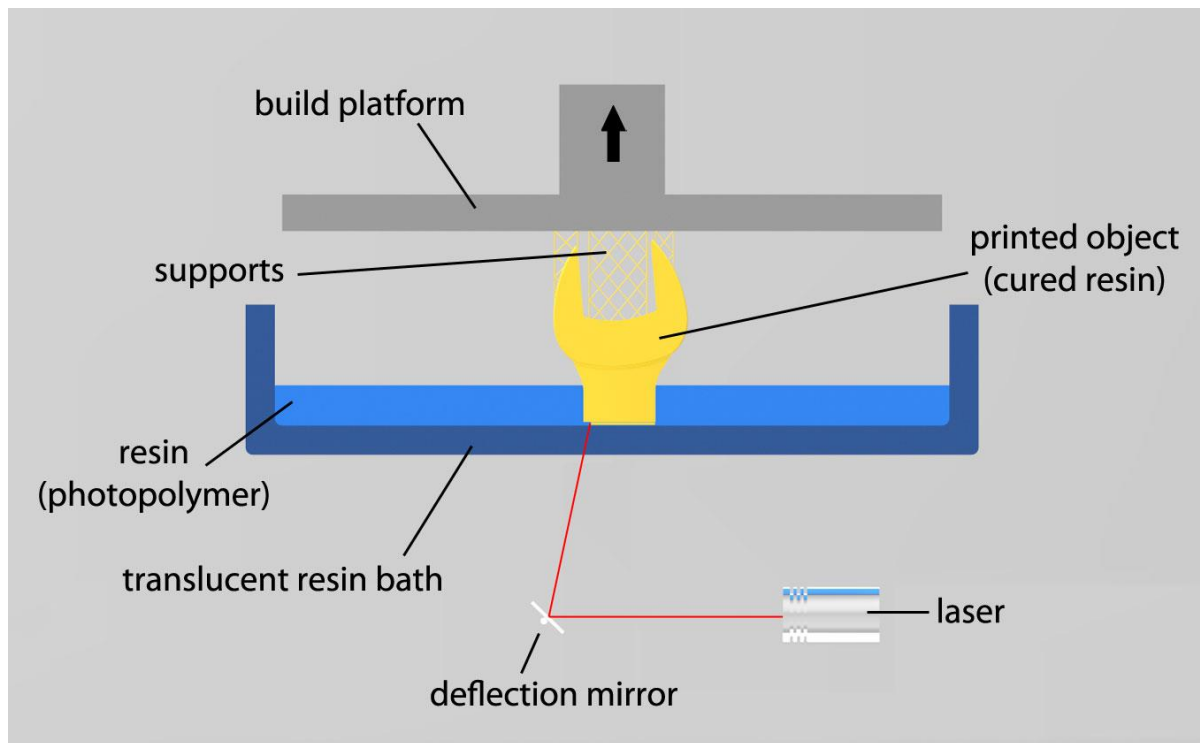


Рисунок 2.18 – SLA друк

Доступні інші форми 3D-принтерів, але по дуже високій ціні. Сюди входять 3D-друк на порошковому шарі (куди падає порошок, який потім зливається у формі), виготовлення ламінованих предметів (склеювання листів матеріалу, а потім їх розрізання за певною формою), спрямоване осадження енергії, а також виготовлення електронно-променевої форми (принтер вистрілює електронний пучок у вакуумі для створення розплавленого металу на основі 3D-моделі).

Ці останні форми 3D-друку зазвичай використовуються для виготовлення металевих деталей, тоді як FDM та SLA найчастіше використовуються для виготовлення пластикових предметів.

Спільним є те, що вони створюють нові об'єкти шляхом поступового додавання та сплавлення сировини.

FDM була першою загальнодоступною технологією 3D-друку для любителів і досі лідує на ринку з точки зору торгових марок та пропозицій товарів, а також кількості проданих одиниць.

Однією з ключових проблем 3D-друку є калібрування налаштувань для успішного друку. Друк може зірватися, оскільки наплавлений пластик нагрівається або охолоджується занадто швидко, або тому що шари не

скріплюються успішно, або тому що модель відривається від поверхні друкувального столу, або тому що в екструдері застряє нитка.

Принтери FDM друкують із широкого спектру пластмас. Кожен пластик має різні характеристики, що може полегшити або ускладнити друк. Також різні пластики дають різні характеристики в готових деталях.

Найпоширенішим типом ниток є PLA. Ним дуже легко друкувати, але він може бути крихким і деформуватися на сонячному світлі.

Нейлон міцний та гнучкий, але часто вимагає багато дрібниць, щоб налаштування друку працювали.

ABS сильніший (саме з нього виготовляються LEGO), але він охолоджується зі швидкістю, яка часто призводить до скручування нижніх шарів, деформуючи всю модель. Він також має неприємний запах і помірно токсичні пари.

Деякі постачальники змішують основний пластик (переважно PLA) з іншими матеріалами, включаючи дерево, метали та вуглецеве волокно. Кожне з них змінює характеристики готового, надрукованого об'єкта.

Більшість принтерів FDM мають один екструдер і можуть друкувати з одного рулону ниток одночасно. Досконаліші (і дорожчі) FDM-принтери можуть друкувати одночасно двома, трьома, чотирьма або більшою кількістю ниток, що дозволяє принтеру поєднувати кольори, функціональні характеристики (наприклад, твердий пластик із гнучкими петлями) та розчинні допоміжні матеріали.

Моделі створюються із струн розплавленого пластику, тому «провисання» можуть стати проблемою. Хоча принтери FDM зазвичай можуть друкувати кола або кути до 45-60 градусів, вони не можуть друкувати через великі повітряні проміжки, оскільки розплавлений пластик просто осяде в щілині.

Щоб компенсувати проблему великих прогалин, більшість принтерів створюють опори або тимчасові пластикові вишки, які можуть утримувати мостові ділянки. Принтери з одинарними екструдерами використовують той самий матеріал, що і сам об'єкт, з різноманітними налаштуваннями, які дозволяють видаляти опори з помірною легкістю.

Принтери SLA мають ряд характеристик, завдяки яким вони дуже сильно відрізняються від інших.

Вони використовують рідку смолу, досить токсичну у своєму незатверділому вигляді. Якщо смола попаде на шкіру, це може спричинити хворобливі опіки або висип.

Готові моделі потрібно обробити у ванні, а потім дати їм затвердіти. Протягом часу затвердіння вони піддаються деформації. Вони також залишаються токсичними.

Завдяки рідкій смолі та обробній ванні, із SLA принтерами працювати набагато складніше, ніж із FDM принтерами.

Принтери SLA часто мають дуже невеликі площі друку, що призводить до крихтих моделей. Смола часто спеціально розроблена для певного принтера, тому користувачі можуть використовувати тільки матеріал від постачальника, що зменшує асортимент.

Незважаючи на це, популярність SLA принтерів почала зростати, здебільшого тому, що вони здатні виробляти моделі з дуже дрібними деталями та малошаровими лініями. Це робить їх особливо придатними для створення прототипів ювелірних конструкцій та форм, невеликих медичних та стоматологічних конструкцій.

Так само, як ви можете використовувати Photoshop для створення графіки, Illustrator для створення ілюстрації або Word для створення статті, програмне забезпечення САПР використовується для створення дизайну 3D-моделі.

Існує безліч програм САПР, кожна з яких найкраще підходить для різних завдань. Найпопулярніші з них – це TinkerCAD та Fusion 360.

TinkerCAD – це дуже проста у використанні програма, яку часто вивчають школярі. Це дозволяє надшвидко створювати прототипи простих конструкцій. Fusion 360 – це повна інженерна проектна програма з функціями не тільки для проектування, але також моделювання руху та стрес-тестування. Є багато інших інструментів, таких як ZBrush та Meshmixer, які часто використовуються для ліплення у віртуальному просторі.

Якщо ви можете намалювати прямокутник у PowerPoint, ви можете використовувати програму САПР для створення простих конструкцій. Ресурси для вивчення загальних програм 3D-друку доступні в багатьох онлайн-курсах, викладаються в коледжах і знаходяться в великому об'ємі на YouTube.

Тим не менш, оскільки інструменти, такі як Fusion 360, можуть бути використані для проектування та віртуального тестування таких проектів, як автомобільні двигуни, їх важко освоїти. Часто необхідна інженерна майстерність, щоб зрозуміти не тільки те, як працює інструмент, але і фізику, що бере участь у роботі кінцевого об'єкта.

Програми САПР створюють віртуальні моделі тривимірних об'єктів. Але зазвичай 3D-друк відбувається пошарово, фрагментами. Процес перетворення 3D-дизайну в серію рухів машини на двовимірній площині (а потім переміщення площини) є завданням програми «розрізання», або «слайсеру».

Більшість слайсерів виробляють G-код, стандартну форму мови числового управління, зрозумілу більшості автоматизованих промислових пристроїв (а не лише 3D-принтерів). Незважаючи на те, що G-код є стандартом, постачальники часто додають розширення та модифікації. Це означає, що G-код, як правило, потрібно генерувати слайсером для певних марок і моделей пристроїв з цифровим управлінням.

Хоча деякими слайсерами можна керувати програмно, просто передавши в них файл 3D-моделі та отримавши вихідний код G-коду, більшість слайсерів сьогодні мають повністю інтерактивний інтерфейс. Це дозволяє оператору регулювати орієнтацію друку та вивчати процес друку пошарово, щоб виявити потенційні проблеми друку перед тим, як друк надсилається на принтер.

У цей же час також налаштовуються різні параметри друку, починаючи від температури сопла та збірки, техніки зчеплення, методів заповнення, швидкості друку та навіть користувацьких блоків G-коду для врахування спеціальних процедур, таких як зупинка друку для вбудовування магнітів.

Як і у випадку з 3D-принтерами та програмами САПР, на вибір є безліч слайсерів. Деякі з найпопулярніших, таких як Cura та Slic3r, є відкритими. Є також надійні комерційні програми, такі як Simplify3D.

Висновки до розділу

В даному розділі було розглянуто основні технології та рішення для розробки роботизованої рухомої платформи. Було розглянуто основні види кінематики роботів, платформи для здійснення різних видів кінематики, платформи для керування складними роботизованими системами та визначено їх переваги і недоліки.

У результаті було прийняте рішення використовувати шестиколесну платформу для найбільшої стабільності, прохідності системи на різних поверхнях, а також для значного запасу вантажопідйомності. Як основна керуюча плата буде використовуватися Raspberry Pi model B, у якості керуючої плати для моторів буде використовуватися окремий драйвер моторів із окремою лінією живлення.

Завдяки великій кількості GPIO виходів та запасу обчислювальної потужності центрального процесора, Raspberry Pi Model B зможе забезпечити можливість зчитування, аналізу та передачі інформації не використовуючи додаткові апаратні рішення, що ідеально підходить під задачі проекту.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ

3.1. Розробка 3D-моделі

Для розробки 3D-моделі прототипу було обрано програму Fusion 360. Fusion 360 – це хмарна платформа САПР, яка є доступною, високопродуктивною альтернативою іншим основним гравцям галузі. Цей додаток простий у використанні та має всі загальні функції, яких можна очікувати від популярних пакетів САПР. Fusion 360 був побудований з нуля, щоб бути всеохоплюючим рішенням для розробки продуктів і може запропонувати простий робочий процес, починаючи від концептуального дизайну і закінчуючи виробництвом.

Fusion 360 має дуже велику базу знань, яка ретельно охоплює всі особливості програмного забезпечення; ці навчальні посібники можна отримати через Fusion 360, а також через веб-сайт Autodesk. Існує також офіційний канал YouTube із багатьма годинами безкоштовних підручників.

Програмне забезпечення часто оновлюється, а нові функції надходять кожні кілька місяців. Fusion 360 ідеально підходить для підприємств з високим товарообігом, а також для стартапів, які шукають професійний інструмент для виведення їх на ринок.

Fusion 360 може виконувати ресурсоємні операції в хмарі, включаючи візуалізацію, моделювання, оптимізацію фігури та генеративний дизайн. Це означає, що робота може тривати, поки весь важкий підйом робиться на хмарі.

У Fusion 360 ви можете перемикатися між шістьма різними робочими областями. Кожна робоча область має власний набір інструментів та функцій:

- Design: Для малювання 3D-моделей та поверхонь за допомогою ескізів, видавлень, обертів та багатьох інших стандартних інструментів САПР.
- Render: Створюйте фотореалістичні візуалізації компонентів та продуктів.
- Animation: Анімуйте збірки, щоб побачити, чи функціонують вони належним чином, або показати функціональність потенційним клієнтам.

- **Simulation:** автоматизована інженерія для проведення різних аналізів напружень на конструкціях, щоб переконатися, що вони здатні відповідати умовам експлуатації.
- **Manufacture:** автоматизоване виробництво (CAM) для надання допомоги у виготовленні деталі на різних цифрових інструментах виготовлення, таких як фрези з ЧПУ, токарні верстати з ЧПУ, лазерні різакі та гідроабразивні різакі.
- **Drawing:** Створіть креслення конструкцій для виготовлення на ЧПУ.

Створення 3D-моделі складається з кількох простих етапів. Перший – створення ескізу.

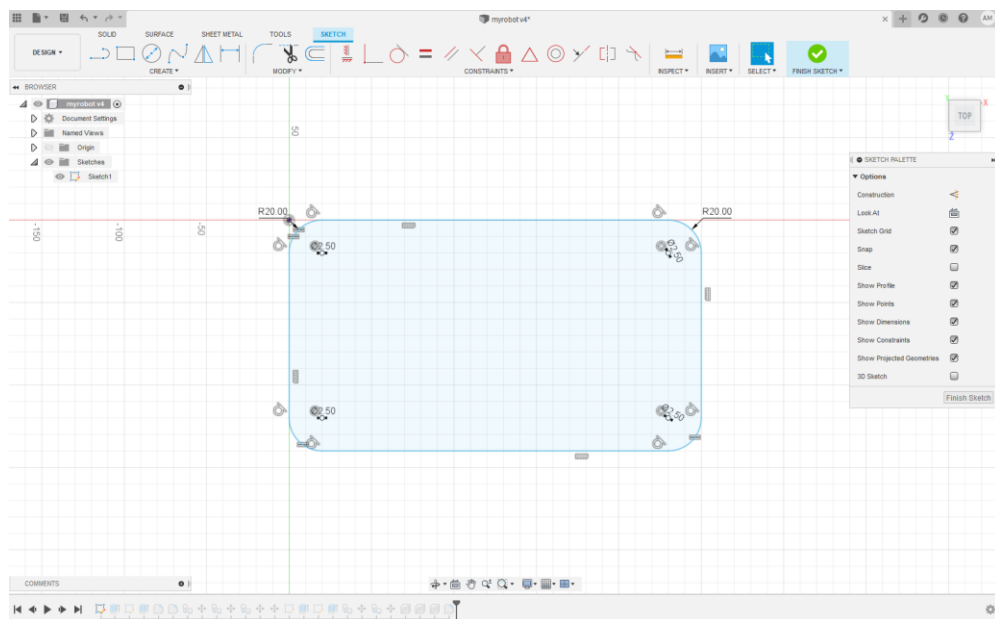


Рисунок 3.1 – Створення ескізу у Fusion 360

Ескізи представляють собою лінії та криві на 2D-площині. Для його створення натисніть кнопку «Створити ескіз» на панелі інструментів і виберіть, на якій площині ви хочете намалювати. Вигляд переключиться з тривимірної на 2D-перспективу.

Другий етап – видавлювання. Щоб розпочати операцію, натисніть кнопку “Extrude” на панелі інструментів (або “E” на клавіатурі). Це має призвести до того, що меню з’явиться в правій частині екрана.

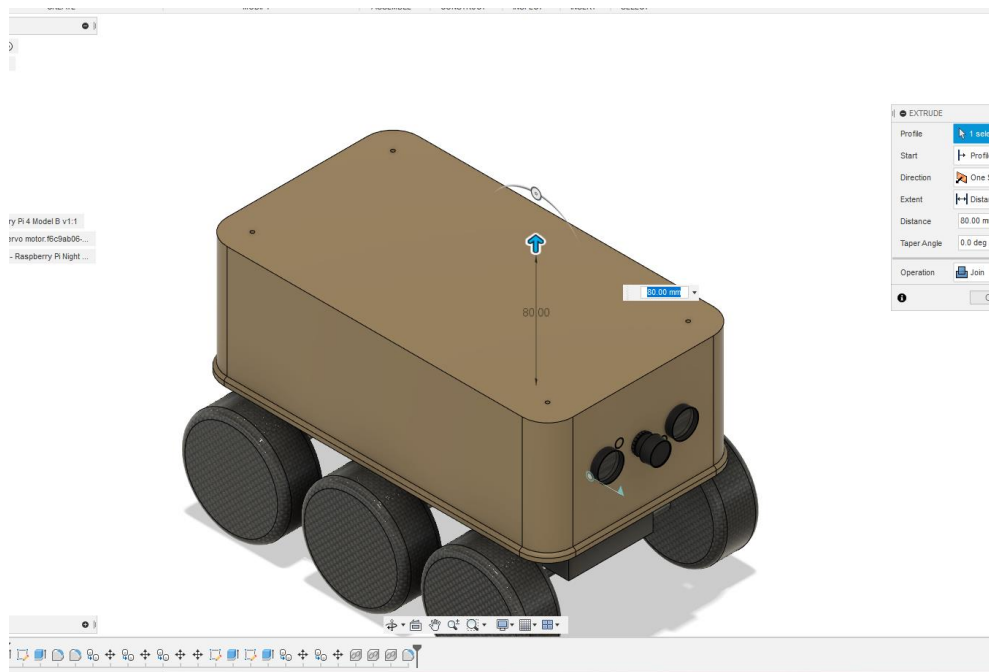


Рисунок 3.2 – Операція видавлювання у Fusion 360

Після цього достатньо застосувати деякі косметичні зміни у об'єкті на експортувати модель. Після цього її можна передати у програму для перетворення моделі на GCODE-файл. Він використовується 3Д-принтером при друці.

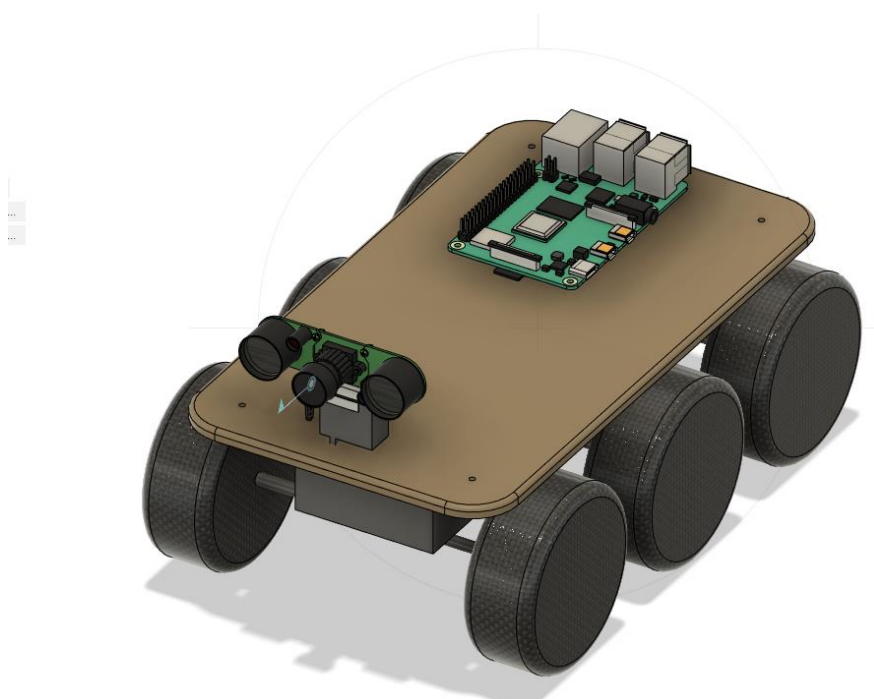


Рисунок 3.3 – 3D модель у Fusion 360

3.2 Архітектура системи

Трьома основними компонентами системи є патрульний робот, сервер і клієнт.

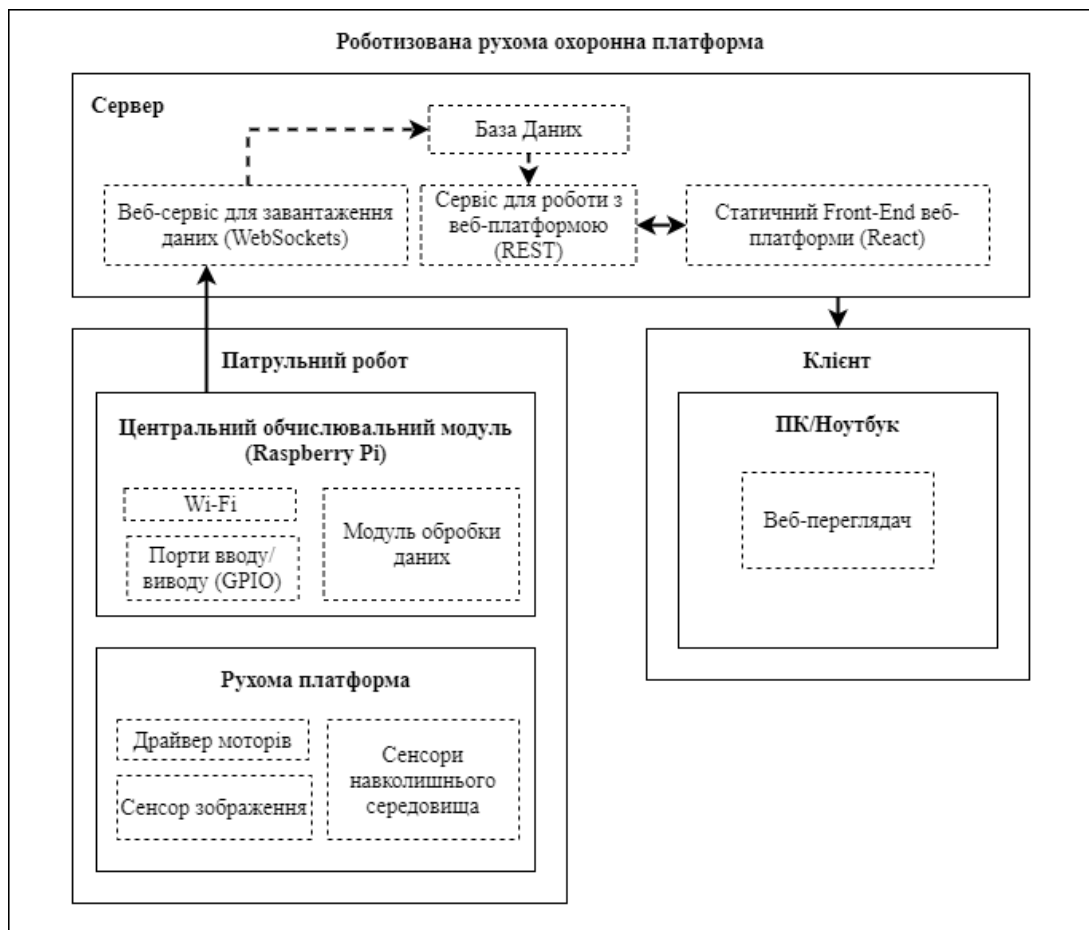


Рисунок 3.4 – Архітектура системи

Центральним обчислювальним модулем РРОП буде виступати Raspberry Pi 4. Усі сенсори та додаткові пристрої будуть підключені до цієї плати. Першим і одним з найголовніших модулів буде модуль камери.



Рисунок 3.5 – Модуль камери

Модуль камери оснащений двома інтенсивними інфрачервоними світлодіодними прожекторами для нічного запису. Інфрачервоні світлодіоди живляться безпосередньо від порту CSI і здатні освітлювати територію на відстані до 8м. Камера також має регульовану фокусну відстань 3,6 мм і кут огляду 75,7 градуса.

ІЧ-світлодіоди можна прикрутити до корпусу камери за допомогою доданих в комплекті болтів. Це підключає живлення 3,3 В до плат і фіксує їх на місці. Якщо потрібно налаштувати поріг навколишнього освітлення (щоб змінити, коли інфрачервоний світлодіод вмикається), на кожній платі світлодіода є крихітний регульований потенціометр, можна повернути його, щоб змінити поріг.

Другим необхідним модулем є модуль контролю температури, вологості і тиску. Таким модулем було обрано BME280.

Пристрій BME280 є цифровим барометричним датчиком тиску і є модернізованою версією BMP180. Датчик умістився на дуже малій платі, яка забезпечує доступ до датчика через інтерфейс I2C. Це дозволяє легко підключити його до Raspberry Pi і читати дані за допомогою Python. BME280 забезпечує контроль температури, тиску і вологості.

BME280 виготовлений компанією Bosch, а офіційна специфікація BME280 включає всі технічні дані [де633401](#)

талі. Їх пристрій може пропонувати як інтерфейси SPI, так і I2C, тому вам потрібно переконатися, що ваш модуль використовує бажаний інтерфейс.

Шина I2C (Inter-IC) – це двонаправлена двопровідна послідовна шина, яка забезпечує комунікаційний зв'язок між інтегральними схемами (IC). Філіпс представив шину I2C 20 років тому для серійних виробів, таких як телевізори, відеомагнітофони та аудіообладнання. Сьогодні I2C є фактичним рішенням для вбудованих додатків.

Існує три швидкості передачі даних для шини I2C: стандартний, швидкий режим і високошвидкісний режим. Стандартний – 100 Кбіт / с. Швидкий режим – 400 Кбіт / с, а високошвидкісний режим підтримує швидкість до 3,4 Мбіт / с.

Всі вони сумісні один з одним. Шина I2C підтримує 7-розрядні та 10-розрядні пристрої адресного простору та пристрої, які працюють під різною напругою.

Для використання VME280 потрібно ввімкнути інтерфейс I2C на Raspberry Pi, оскільки він не ввімкнений за замовчуванням. Це досить простий процес.

Спочатку треба запустити `sudo raspi-config` і дотримуватися вказівок, щоб встановити підтримку i2c для ядра ARM та ядра Linux.

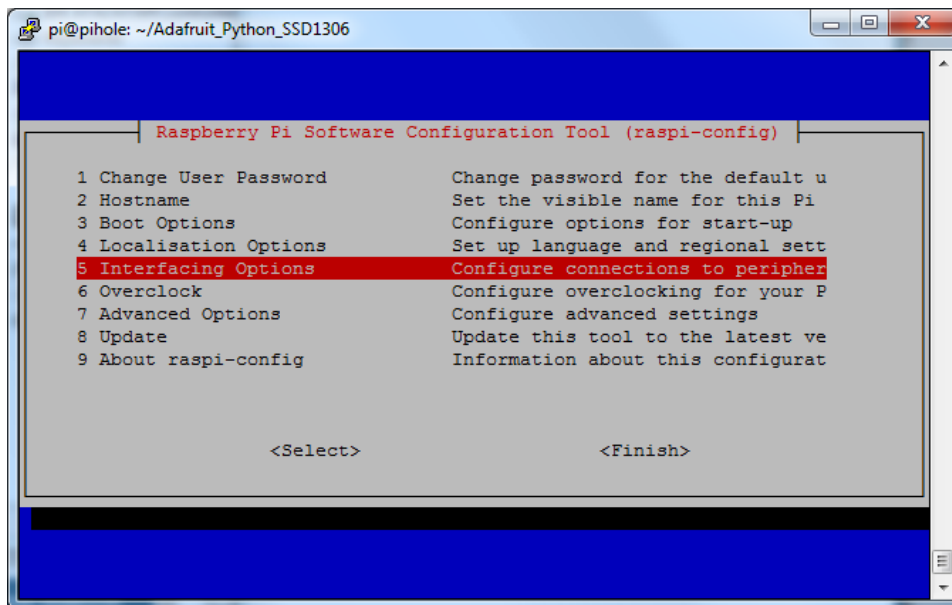


Рисунок 3.6 – Перехід до пункту «Interfacing options»

Для того, щоб протестувати роботу I2C – можна ввести команду `sudo i2cdetect -y 1`.

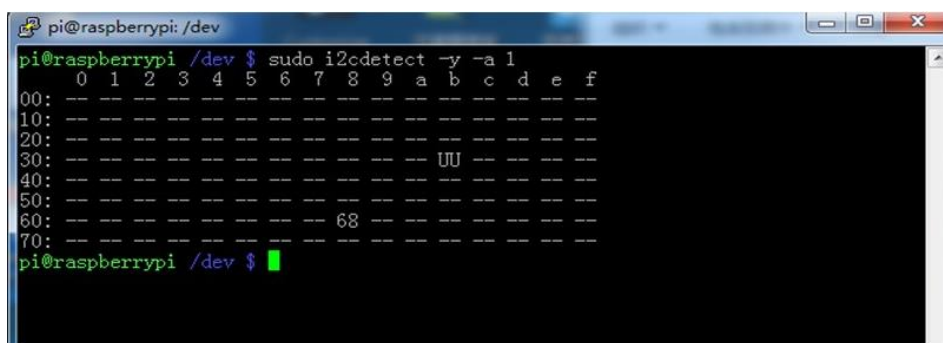


Рисунок 3.7 – Тестування роботи I2C

На рис. 3.7. видно, що використовуються дві адреси I2C – `bх30` та `8х60`.

Ці значення будуть різними залежно від того, що в даний час приєднано до роз'ємів I2C RPi.

Інтерфейс I2C досить простий (рис. 3.8). Він має лише чотири дроти: SCL (послідовний годинник), SDA (послідовні дані), GND (земля) та VIN (вхід

напруги). Адреса I2C датчика зазвичай визначається виробником. Плати одного типу, виготовлені різними виробниками, зазвичай мають однакову адресу для забезпечення сумісності.

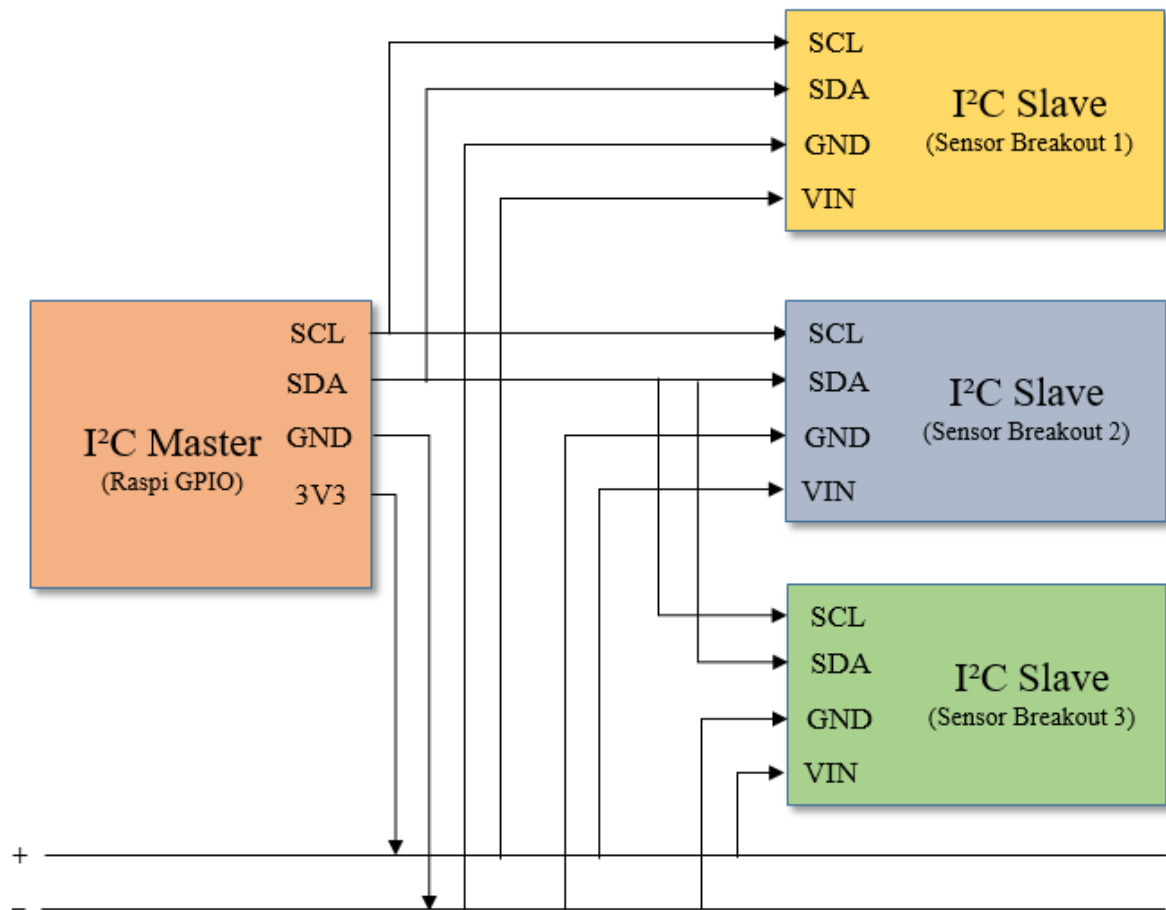


Рисунок 3.8 – Схема шини I2C, яка з'єднує декілька сенсорів

Багато з сенсорів мають драйвери з відкритим кодом, написані на Python та Java, доступні в Raspberry Pi. Тож ви можете легко розпочати роботу, якщо ви вже знайомі з цими мовами програмування.

Єдина незручність цих плат полягає в тому, що користувачеві доводиться самостійно займатися пайкою (дуже мало плат постачаються з вже запаяними конекторами). Але як тільки користувач пройде цей крок, все повинно бути відносно легким. Давайте подивимося, що деякі з цих проривів датчиків можуть зробити для нас.

Розглянемо схему підключення BME280 до RaspberryPi.

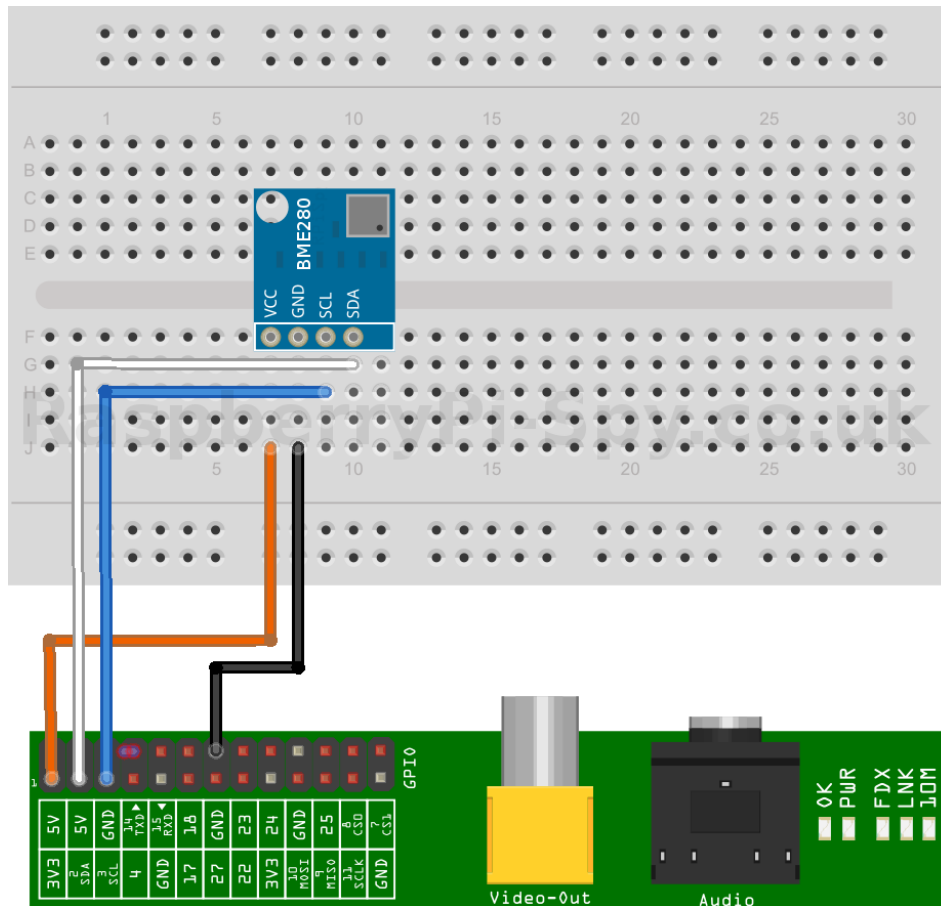


Рисунок 3.9 – Схема підключення BME280 до Raspberry Pi

В якості сенсора світла застосуємо звичайний фоторезистор (рис. 3.10).



Рисунок 3.10 – Фоторезистор

Світлозалежний резистор або також відомий як датчик LDR у світлій кімнаті матиме опір лише кілька сотень Ом, а в темряві може мати опір у кілька мегаомів.

Для того, щоб виміряти опір фоторезистора ми застосуємо звичайний конденсатор (рис 3.9). Конденсатор по суті діє як акумулятор, заряджаючись під час отримання енергії, а потім розряджаючись, коли більше не отримує

живлення. Використовуючи його послідовно з фоторезистором, ми можемо визначити, скільки опору надає фоторезистор, таким чином маючи змогу визначити, наскільки світло чи темно в приміщенні.



Рисунок 3.11 – Конденсатор на 1мкФ

Щоб правильно під'єднати фоторезистор до RaspberryPi при прототипуванні, треба виконати наведені нижче дії (рис. 3.12).

- Підключити GPIO №1 (3v3) до позитивної шини (+) на макетній дошці.
- Підключити GPIO №6 (земля) до шини заземлення (-) на макетній дошці.
- Помістити фоторезистор на плату і під'єднати провід від одного з контактів до позитивного контакту конденсатора.
- До іншого контакту фоторезистора підключити дріт, що веде назад до Raspberry Pi, до GPIO №7.
- Під'єднати негативний контакт конденсатора до шини заземлення на макетній дошці.

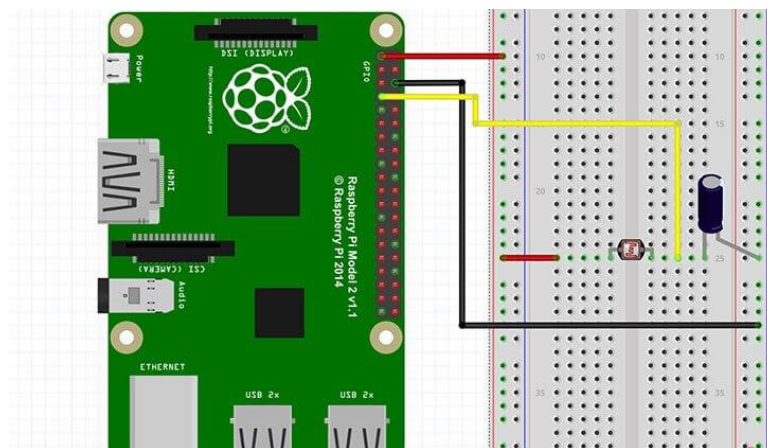


Рисунок 3.12 – Схема під'єднання фоторезистора до Raspberry Pi

Найбільша проблема, з якою зіткнувся із цією схемою, полягає в тому, що RPi не має жодного аналогового контакту. Усі вони цифрові, тому ми не можемо точно виміряти опір на вході.

Натомість ми виміряємо час, який потрібно конденсатору, щоб зарядитися і надіслати на контакт HIGH сигнал. Цей метод – простий, але неточний спосіб визначити освітленість.

Наведемо лістинг коду для зчитування інформації з датчика освітленості на Raspberry Pi. Для початку ми імпортуємо пакет GPIO, який нам знадобиться, щоб ми могли зв'язатися із контактами GPIO.

Також імпортуємо пакет time, що ми могли перевести сценарій у режим сну, коли нам потрібно.

```
#!/usr/local/bin/python

import RPi.GPIO as GPIO
import time
```

Рисунок 3.13 – імпорт пакету time

Потім ми встановлюємо режим GPIO на GPIO.BOARD, і це означає, що вся нумерація, яку ми використовуємо в цьому сценарії, буде стосуватися фізичної нумерації контактів.

Оскільки ми використовуємо лише один контакт на вхід/вихід, нам потрібно оголосити лише одну змінну. Встановлюємо для цієї змінної номер контакту, який використовуємо як вхідний / вихідний.

```
GPIO.setmode(GPIO.BOARD)

#define the pin that goes to the circuit
pin_to_circuit = 7
```

Рисунок 3.14 – Задання режиму роботи GPIO

Далі створюємо функцію, яка називається `rc_time`, і яка вимагає одного параметра, який є номером контакту на платі. У цій функції ми ініціалізуємо змінну, яка називається `count`, і повернемо значення цієї змінної, коли контакт перейде у HIGH.

Потім ми встановлюємо наш контакт, як вихідний і встановлюємо його значення LOW. Далі виконується сценарій сну на 10мс.

Після цього встановлюємо контакт, як вхідний і вводимо цикл while. Залишаємось у цьому циклі поки контакт не перейде у значення HIGH (коли конденсатор заряджається приблизно до 3/4).

Як тільки контакт переходить у значення HIGH, повертаємо значення підрахунку до основної функції.

```
def rc_time (pin_to_circuit):  
    count = 0  
  
    GPIO.setup(pin_to_circuit, GPIO.OUT)  
    GPIO.output(pin_to_circuit, GPIO.LOW)  
    time.sleep(0.1)  
  
    GPIO.setup(pin_to_circuit, GPIO.IN)  
  
    while (GPIO.input(pin_to_circuit) == GPIO.LOW):  
        count += 1  
  
    return count  
  
try:  
    while True:  
        print(rc_time(pin_to_circuit))  
except KeyboardInterrupt:  
    pass  
finally:  
    GPIO.cleanup()
```

Рисунок 3.15 – Визначення часу переходу стану конденсатора
Системою контролю двигунів виступатиме Adafruit Motor Shield V2.

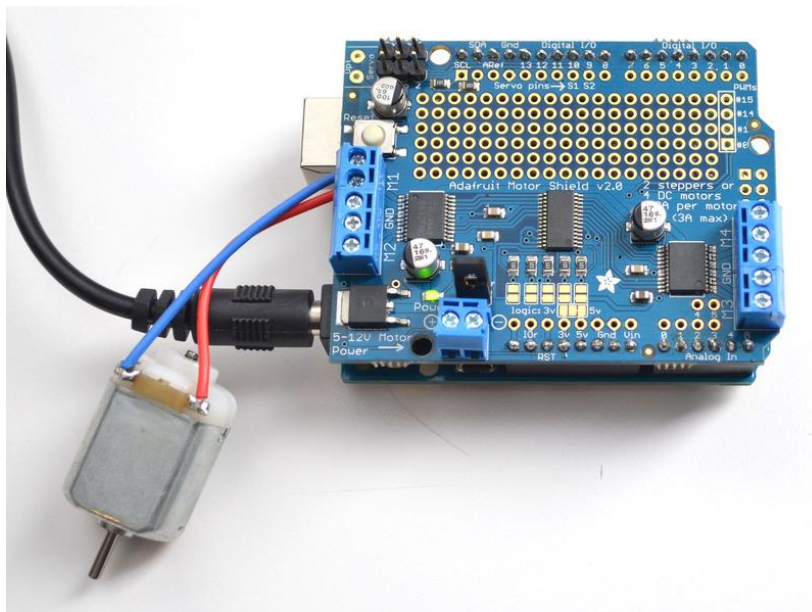


Рисунок 3.16 – Adafruit Motor Shield V2

3.3 Програмне забезпечення компонентів рухомої платформи

Спочатку робот буде шукати прохідний шлях, використовуючи інформацію про глибину сцени з камери. Робот підходить до всіх виявлених пунктів огляду та підписується електронним способом, після чого завантажується номер пункту огляду та час входу. Коли всі пункти огляду пройдено, це конкретне завдання патрулювання буде виконано. На рис. 3.14 представлена блок-схема, що демонструє процес автономного патрулювання.

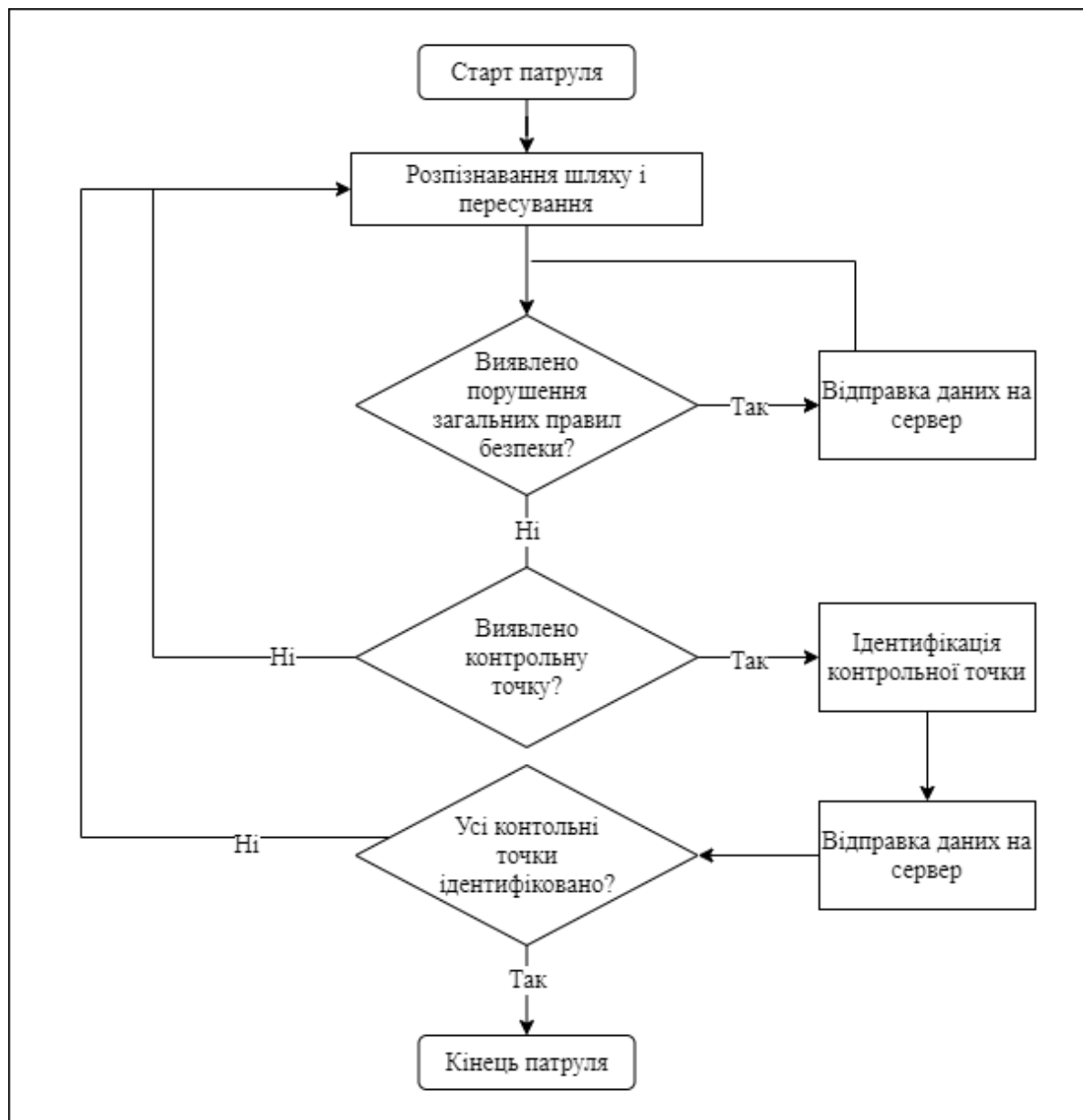


Рисунок 3.17 – Алгоритм патрулювання

На сервері використовується два веб-сервіси (сервіс інформації у реальному часі та сервіс завантаження/відображення збережених даних) для віддаленого резервного копіювання даних та моніторингу. Робот створює

резервну копію інформації під час проходження контрольних точок та при виявленні виключних ситуацій. Центральний модуль RPi підключається до бездротової точки доступу за допомогою модуля Wi-Fi. Уся інформація про стан контрольних точок передається у вигляді JSON-об'єкта, в якому знаходиться інформація про температуру, вологість повітря, тиск, освітлення і шум, а також фото контрольної точки, закодоване у форматі base64. Також робот на постійній основі передає зображення з камери на WebSocket-сервер. Користувач може зайти на веб-платформу та переглянути зображення та інформацію з робота у реальному часі, або дані за певний період часу.

Насамперед, необхідно здійснити підключення моторів до драйвера моторів. Для цього потрібно:

- Підключити два дроти від першого та другого двигуна до клем M1 та M2 на платі.
- Підключити позитивну і негативну клему живлення до позитивного і негативного терміналу плати.

Потім необхідно встановити останню версію Adafruit CircuitPython. Потім для керування двигунами застосовуємо цей код на рис. 3.15.

```
import time
import board
from adafruit_motorkit import MotorKit

kit = MotorKit(i2c=board.I2C())

kit.motor1.throttle = 1.0
time.sleep(0.5)
kit.motor1.throttle = 0
```

Рисунок 3.18 – Код для керування двигунами

Для отримання і збереження інформації з датчика BME280 будемо використовувати Azure IoT Hub. Спочатку створимо новий IoT Hub (рис. 3.15). Для цього необхідно мати аккаунт Microsoft Azure.

Home > New > IoT hub

IoT hub
Microsoft

Basics Size and scale Tags Review + create

Create an IoT Hub to help you connect, monitor, and manage billions of your IoT assets. [Learn more](#)

Project details
Choose the subscription you'll use to manage deployments and costs. Use resource groups like folders to help you organize and manage resources.

Subscription * ⓘ Personal IoT items

Resource group * ⓘ [Create new](#)

Region * ⓘ East Asia

IoT hub name * ⓘ Once your hub is created, this name can't be changed

Review + create < Previous Next: Size and scale > Automation options

Рисунок 3.19 – Створення Microsoft Azure IoT Hub

Далі необхідно додати новий пристрій до IoT Hub. Для цього треба:

- У меню навігації IoT Hub відкрити IoT Devices, а потім вибрати New, щоб додати пристрій у IoT Hub.
- У розділі «Створення пристрою» ввести ім'я нового пристрою, наприклад myDeviceId, і вибрати Зберегти. Ця дія створює ідентифікацію пристрою для IoT Hub.
- Після створення пристрою відкрити його зі списку на панелі пристроїв IoT. Скопіювати рядок основного підключення, щоб використовувати пізніше.

Для зчитування даних з BME280 застосуємо код на рис. 3.17, рис. 3.18 і рис. 3.19.

```
smbus
import time</p><p># Get I2C bus
bus = smbus.SMBus(1)</p><p># BME280 address, 0x76(118)
# Read data back from 0x88(136), 24 bytes
b1 = bus.read_i2c_block_data(0x76, 0x88, 24)</p><p># Convert the data
```

Рисунок 3.20 – Код для зчитування даних з BME280

```

# Temp coefficients
dig_T1 = bl[1] * 256 + bl[0]
dig_T2 = bl[3] * 256 + bl[2]
if dig_T2 > 32767 :
    dig_T2 -= 65536
dig_T3 = bl[5] * 256 + bl[4]
if dig_T3 > 32767 :
    dig_T3 -= 65536</p><p># Pressure coefficients
dig_P1 = bl[7] * 256 + bl[6]
dig_P2 = bl[9] * 256 + bl[8]
if dig_P2 > 32767 :
    dig_P2 -= 65536
dig_P3 = bl[11] * 256 + bl[10]
if dig_P3 > 32767 :
    dig_P3 -= 65536
dig_P4 = bl[13] * 256 + bl[12]
if dig_P4 > 32767 :
    dig_P4 -= 65536
dig_P5 = bl[15] * 256 + bl[14]
if dig_P5 > 32767 :
    dig_P5 -= 65536
dig_P6 = bl[17] * 256 + bl[16]
if dig_P6 > 32767 :
    dig_P6 -= 65536
dig_P7 = bl[19] * 256 + bl[18]
if dig_P7 > 32767 :
    dig_P7 -= 65536
dig_P8 = bl[21] * 256 + bl[20]
if dig_P8 > 32767 :
    dig_P8 -= 65536
dig_P9 = bl[23] * 256 + bl[22]
if dig_P9 > 32767 :
    dig_P9 -= 65536</p><p># BME280 address, 0x76(118)
# Read data back from 0xA1(161), 1 byte
dig_H1 = bus.read_byte_data(0x76, 0xA1)</p><p># BME280 address, 0x76(118)
# Read data back from 0xE1(225), 7 bytes
bl = bus.read_i2c_block_data(0x76, 0xE1, 7)</p><p># Convert the data
# Humidity coefficients
dig_H2 = bl[1] * 256 + bl[0]
if dig_H2 > 32767 :
    dig_H2 -= 65536
dig_H3 = (bl[2] & 0xFF)
dig_H4 = (bl[3] * 16) + (bl[4] & 0xF)
if dig_H4 > 32767 :
    dig_H4 -= 65536
dig_H5 = (bl[4] / 16) + (bl[5] * 16)
if dig_H5 > 32767 :
    dig_H5 -= 65536
dig_H6 = bl[6]
if dig_H6 > 127 :
    dig_H6 -= 256</p><p># BME280 address, 0x76(118)

```

Рисунок 3.21 – Код для зчитування даних з BME280


```

# Select control humidity register, 0xF2(242)
#      0x01(01)      Humidity Oversampling = 1
bus.write_byte_data(0x76, 0xF2, 0x01)
# BME280 address, 0x76(118)
# Select Control measurement register, 0xF4(244)
#      0x27(39)      Pressure and Temperature Oversampling rate = 1
#                      Normal mode
bus.write_byte_data(0x76, 0xF4, 0x27)
# BME280 address, 0x76(118)
# Select Configuration register, 0xF5(245)
#      0xA0(00)      Stand by time = 1000 ms
bus.write_byte_data(0x76, 0xF5, 0xA0)</p><p>time.sleep(0.5)</p><p>
# Read data back from 0xF7(247), 8 bytes
# Pressure MSB, Pressure LSB, Pressure xLSB, Temperature MSB, Temperature LSB
# Temperature xLSB, Humidity MSB, Humidity LSB
data = bus.read_i2c_block_data(0x76, 0xF7, 8)</p><p>
adc_p = ((data[0] * 65536) + (data[1] * 256) + (data[2] & 0xF0)) / 16
adc_t = ((data[3] * 65536) + (data[4] * 256) + (data[5] & 0xF0)) / 16</p><p>
adc_h = data[6] * 256 + data[7]</p><p># Temperature offset calculations
var1 = ((adc_t) / 16384.0 - (dig_T1) / 1024.0) * (dig_T2)
var2 = (((adc_t) / 131072.0 - (dig_T1) / 8192.0) * ((adc_t)/131072.0
        - (dig_T1)/8192.0)) * (dig_T3)
t_fine = (var1 + var2)
cTemp = (var1 + var2) / 5120.0
fTemp = cTemp * 1.8 + 32</p><p># Pressure offset calculations
var1 = (t_fine / 2.0) - 64000.0
var2 = var1 * var1 * (dig_P6) / 32768.0
var2 = var2 + var1 * (dig_P5) * 2.0
var2 = (var2 / 4.0) + ((dig_P4) * 65536.0)
var1 = ((dig_P3) * var1 * var1 / 524288.0 + (dig_P2) * var1) / 524288.0
var1 = (1.0 + var1 / 32768.0) * (dig_P1)
p = 1048576.0 - adc_p
p = (p - (var2 / 4096.0)) * 6250.0 / var1
var1 = (dig_P9) * p * p / 2147483648.0
var2 = p * (dig_P8) / 32768.0
pressure = (p + (var1 + var2 + (dig_P7))) / 16.0) / 100</p><p>
var_H = ((t_fine) - 76800.0)
var_H = (adc_h - (dig_H4 * 64.0 + dig_H5 / 16384.0 * var_H)) * (dig_H2 / 65536.0
        * (1.0 + dig_H6 / 67108864.0 * var_H * (1.0 + dig_H3 / 67108864.0 * var_H)))
humidity = var_H * (1.0 - dig_H1 * var_H / 524288.0)
if humidity > 100.0 :
    humidity = 100.0
elif humidity < 0.0 :
    humidity = 0.0</p><p># Output data to screen
print "Temperature in Celsius : %.2f C" %cTemp
print "Temperature in Fahrenheit : %.2f F" %fTemp
print "Pressure : %.2f hPa " %pressure
print "Relative Humidity : %.2f %" %humidity</p>

```

Рисунок 3.22 – Код для зчитування даних з BME280

Як платформу для руху системи було обрано платформу Wild Thumper.

Кожен із шести моторів, прикріплених до Wild Thumper, управляє одним колесом. Ці мотори мають такі властивості при напрузі 7 В: 130 обертів на хвилину, 450 мА струм без навантаження і 0,96 Н м крутного моменту.



Рисунок 3.23 – Мотор Wild Thumper

Два біти, або цифрові сигнали, використовуються для визначення напрямку роботи двигуна. У табл. 3.1 пояснюється, як можливі комбінації цих бітів визначатимуть напрямок обертання двигуна.

Таблиця 3.1 – Комбінації управляючих бітів на мотор

Біт 1	Біт 2	Напрямок обертання
0	0	Не обертається
0	1	Назад
1	0	Вперед
1	1	Не обертається

В залежності від положення двигуна користувач може вибрати, в якому напрямку будуть відбуватися рухи "вперед" і "назад". Логічне значення цих бітів, наприклад LOW або HIGH, визначає напрямок обертання двигунів; а отже і колес, прикріплених до них. Потужність джерела напруги визначає швидкість обертання двигунів. Під час використання картографічного робота висока швидкість не потрібна. Більш низькі напруги призводять до лінійно менших швидкостей обертання в хвилину. Найнижчий темп, необхідний для управління двигунами, становить приблизно 60 об / хв.

3.4 Програмне забезпечення веб-платформи

Програмне забезпечення платформи буде базуватися на технологіях NodeJS + Express та React. Node – це середовище JavaScript, побудоване на тому самому механізмі JavaScript, який використовується у веб-браузері Google Chrome. Він має кілька чудових функцій, які роблять його привабливим

вибором для створення додатків середнього рівня на стороні сервера, включаючи веб-сервери та веб-служби для API платформ.

Модель вводу-виводу, керована подіями, що не блокуються, надає їй дуже привабливу продуктивність, легко перевершуючи потокові серверні середовища, такі як PHP та Ruby on Rails, які блокують введення / виведення та обробляють одночасно запити декількох користувачів, використовуючи окремі потоки для кожного.

Деякі особливості NodeJS:

- Швидка робота (Неблокуючий ввід / вивід за замовчуванням).
- Події
- Швидкий обмін даними.
- Добре спроектований потоковий API.
- Зручні стандартні бібліотеки для взаємодії з ОС, файловою системою тощо.
- Підтримка скомпільованих двійкових модулів на випадок, коли вам потрібно розширити можливості Node за допомогою мови нижчого рівня, такої як C ++.
- NodeJS довіряють і підтримують великі підприємства, що використовують критично важливі програми. (Adobe, Google, Microsoft, Netflix, PayPal, Uber, Walmart тощо.).
- Низький поріг входу.

Для того, щоб запустити найпростіший сервер на NodeJS достатньо коду на рис. 3.21.

```
const express = require('express');

const app = express();
const port = process.env.PORT || 3000;

app.get('/', (req, res) => {
  res.send('\n\nHello, world!\n\n');
});

app.listen(port, () => {
  console.log(`listening on port ${ port }`);
});
```

Рисунок 3.24 – Код для запуску сервера NodeJS

Express.js – це фреймворк веб-додатків, який побудований поверх Node.js. Він забезпечує мінімальний інтерфейс з усіма необхідними інструментами, необхідними для створення веб-додатків. Express.js покращує гнучкість додатку за допомогою великого набору модулів, доступних на npm, які можна безпосередньо підключити до Express. Це допомагає легко керувати потоком даних між сервером і маршрутами (routes) в серверних додатках. Він був розроблений TJ Holowaychuk і випущений на ринок 22 травня 2010 року. Раніше ним керував IBM, але в даний час він знаходиться під керівництвом інкубатора Node.js Foundation.

Express в основному відповідає за обробку внутрішньої частини в стеку MEAN. MEAN Stack – це програмний стек з відкритим кодом JavaScript, який активно використовується для створення динамічних веб-сайтів та веб-додатків на ринку. Тут MEAN розшифровується як MongoDB, Express.js, та Node.js.

Для роботи із користувачем нам спочатку необхідно описати MongoDB модель. На рис. 3.22 наведено код для найпростішої моделі користувача:

```
const mongoose = require('mongoose');
const UserSchema = new mongoose.Schema({
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true }
});
module.exports = mongoose.model('User', UserSchema);
```

Рисунок 3.25 – Найпростіша модель користувача

Цей код дозволяє створювати об'єкти користувачів, які мають власні унікальні поля електронної пошти та пароля, які потім можна зберегти та отримати з MongoDB для аутентифікації користувачів.

Паролі не можна зберігати в простому тексті, тому для захисту паролів використовується невелика бібліотека bcrypt, це дозволить хешувати паролі.

Після встановлення bcrypt можна додати в схему користувача hook для хешування паролів, перш ніж зберігати їх у базі даних:

```

const mongoose = require('mongoose');
const bcrypt = require('bcrypt');
const saltRounds = 10;
const UserSchema = new mongoose.Schema({
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true }
});
UserSchema.pre('save', function(next) {
  if (this.isNew || this.isModified('password')) {
    const document = this;
    bcrypt.hash(document.password, saltRounds,
      function(err, hashedPassword) {
        if (err) {
          next(err);
        }
        else {
          document.password = hashedPassword;
          next();
        }
      });
  }
  next();
});
module.exports = mongoose.model('User', UserSchema);

```

Рисунок 3.26 – Хешування паролів

Тепер, коли об'єкт User створено, можна перевірити його та створити деяких користувачів. Для цього створюємо ще один Express-маршрут так:

```

app.post('/api/register', function(req, res) {
  const { email, password } = req.body;
  const user = new User({ email, password });
  user.save(function(err) {
    if (err) {
      res.status(500)
        .send("Error registering new user please try again.");
    } else {
      res.status(200).send("Welcome to the club!");
    }
  });
});

```

Рисунок 3.27 – Маршрут реєстрації користувачів

Тепер можна перевірити цей ендпоїнт за допомогою cURL запису:

```

curl -X POST \
  http://localhost:3000/api/register \
  -H 'Content-Type: application/json' \
  -d '{
    "email": "me@example.com",
    "password": "mypassword"
  }'

```

Рисунок 3.28 – Перевірка ендпоїнта

Тепер, коли декілька користувачів збережено у базі даних, необхідно створити спосіб їх аутентифікації використовуючи базу даних.

Для додаємо метод до схеми користувача, який прийме пароль у вигляді рядка, і за допомогою bcrypt повідомить, чи це правильний пароль для цього користувача:

```
UserSchema.methods.isCorrectPassword = function(password, callback){
  bcrypt.compare(password, this.password, function(err, same) {
    if (err) {
      callback(err);
    } else {
      callback(err, same);
    }
  });
}
```

Рисунок 3.29 – Метод перевірки правильності пароля

Тепер, можна почати видавати токени клієнтам. Спочатку треба створити секретний рядок, який слід використовувати під час підписання токенів.

Далі потрібно встановити бібліотеку jsonwebtoken, яка дозволить видавати та перевіряти веб-маркери JSON.

Нарешті, можна створити новий Express-маршрут, який, отримавши електронну адресу та пароль, знайде користувача із вказаною адресою електронної пошти та переконається, що вказаний пароль правильний. Якщо пароль правильний, ми видамо запитуваний токен користувачу:

```

const jwt = require('jsonwebtoken');
app.post('/api/authenticate', function(req, res) {
  const { email, password } = req.body;
  User.findOne({ email }, function(err, user) {
    if (err) {
      console.error(err);
      res.status(500)
        .json({
          error: 'Internal error please try again'
        });
    } else if (!user) {
      res.status(401)
        .json({
          error: 'Incorrect email or password'
        });
    } else {
      user.isCorrectPassword(password, function(err, same) {
        if (err) {
          res.status(500)
            .json({
              error: 'Internal error please try again'
            });
        } else if (!same) {
          res.status(401)
            .json({
              error: 'Incorrect email or password'
            });
        } else {
          // Issue token
          const payload = { email };
          const token = jwt.sign(payload, secret, {
            expiresIn: '1h'
          });
          res.cookie('token', token, { httpOnly: true })
            .sendStatus(200);
        }
      });
    }
  });
});
});
});

```

Рисунок 3.30 – Видача токена користувачу

Тепер, було створено спосіб видавати підписаний токен уповноваженим користувачам, потрібно визначити, які маршрути в додатку не підходять для неавторизованих користувачів. У цьому випадку необхідно, щоб певний маршрут був доступний лише за умови, що клієнт-запитувач має дійсний токен.

По-перше, треба переконатися, що у на сервері встановлений пакет `cookie-parser`, щоб можна було реалізувати можливість аналізу файлів `cookie`, переданих браузером.

Після цього додамо `middleware cookie-parser` до конфігурації Express:


```
// server.js
const cookieParser = require('cookie-parser');
...
app.use(cookieParser());
```

Рисунок 3.31 – Ініціалізація cookie-parser

Тепер можна створити власний Express middleware, який розміщуватиметься між запитом та захищеним маршрутом та перевірятиме, чи є запит авторизованим. Middleware буде шукати токен у cookie із запиту, а потім перевіряти його.

```
// middleware.js
const jwt = require('jsonwebtoken');
const secret = 'mysecretssshhh';
const withAuth = function(req, res, next) {
  const token = req.cookies.token;
  if (!token) {
    res.status(401).send('Unauthorized: No token provided');
  } else {
    jwt.verify(token, secret, function(err, decoded) {
      if (err) {
        res.status(401).send('Unauthorized: Invalid token');
      } else {
        req.email = decoded.email;
        next();
      }
    });
  }
}
module.exports = withAuth;
```

Рисунок 3.32 – Створення Express middleware

Нарешті, можна використовувати цей middleware, коли є необхідність захистити маршрут. Достатньо просто відредагувати його конфігурацію, щоб використовувати новий middleware:

```
// server.js
const withAuth = require('./middleware');
...
app.get('/api/secret', withAuth, function(req, res) {
  res.send('The password is potato');
});
```

Рисунок 3.33 – Використання middleware у маршрутах Express

Пізніше стане в нагоді спосіб просто запитати сервер, чи є дійсним маркер, збережений у файлах cookie браузера.

Для цього достатньо створити простий маршрут, який поверне статус HTTP 200, якщо користувач має дійсний маркер:


```
app.get('/checkToken', withAuth, function(req, res) {
  res.sendStatus(200);
});
```

Рисунок 3.34 – Метод перевірки токена

Після створення серверної частини, яка може реєструвати та аутентифікувати користувачів, можна працювати над front-end частиною додатку. Для початку створимо простий компонент React з формою, яка буде використовуватися для аутентифікації користувача:

```
import React, { Component } from 'react';
export default class Login extends Component {
  constructor(props) {
    super(props)
    this.state = {
      email: '',
      password: ''
    };
  }
  handleInputChange = (event) => {
    const { value, name } = event.target;
    this.setState({
      [name]: value
    });
  }
  onSubmit = (event) => {
    event.preventDefault();
    alert('Authentication coming soon!');
  }
  render() {
    return (
      <form onSubmit={this.onSubmit}>
        <h1>Login Below!</h1>
        <input
          type="email"
          name="email"
          placeholder="Enter email"
          value={this.state.email}
          onChange={this.handleInputChange}
          required
        />
        <input
          type="password"
          name="password"
          placeholder="Enter password"
          value={this.state.password}
          onChange={this.handleInputChange}
          required
        />
        <input type="submit" value="Submit"/>
      </form>
    );
  }
}
```

Рисунок 3.35 – Форма авторизації користувача

Наш експрес-додаток встановлює для нас cookie httpOnly із підписаним JSON веб-токеном, тому все, що потрібно зробити для аутентифікації, це правильно перенаправити користувача, якщо він отримає статус код 200 HTTP під час виклику методу аутентифікації із додатку React:

```
onSubmit = (event) => {
  event.preventDefault();
  fetch('/api/authenticate', {
    method: 'POST',
    body: JSON.stringify(this.state),
    headers: {
      'Content-Type': 'application/json'
    }
  })
  .then(res => {
    if (res.status === 200) {
      this.props.history.push('/');
    } else {
      const error = new Error(res.error);
      throw error;
    }
  })
  .catch(err => {
    console.error(err);
    alert('Error logging in please try again');
  });
}
```

Рисунок 3.36 – Переадресація користувача після авторизації

Тепер можна додати компонент Login до конфігурації маршруту:

```
import Login from './Login';
...
<Route path="/login" component={Login} />
```

Рисунок 3.37 – Додання компоненту до конфігурації маршруту

Отже, налаштовано процес аутентифікації, який отримує підписаний токен із сервера, зберігає його в cookie і згодом використовує цей маркер для доступу до захищених маршрутів на сервері.

Наостанок, необхідний спосіб вказати маршрути для захисту у інтерфейсі, щоб користувач не бачив компонент, якщо він не увійшов у систему. Натомість необхідно перенаправити його на сторінку входу, як це слід робити в добре спроектованому веб-додатку.

Для цього використовується концепція, яка називається компонентами вищого порядку, щоб обернути маршрути компонентів, які необхідно захистити.

Компонент вищого порядку – це не що інше, як функція, яка приймає компонент і повертає компонент. Отже, необхідно створити компонент вищого порядку, `withAuth`, який буде приймати компонент, який необхідно захистити, наприклад `<Secret />`, і трохи модифікувати його, щоб користувачі не мали до нього доступу, якщо вони не ввійшли в систему:

```
import React, { Component } from 'react';
import { Redirect } from 'react-router-dom';
export default function withAuth(ComponentToProtect) {
  return class extends Component {
    constructor() {
      super();
      this.state = {
        loading: true,
        redirect: false,
      };
    }
    componentDidMount() {
      fetch('/checkToken')
        .then(res => {
          if (res.status === 200) {
            this.setState({ loading: false });
          } else {
            const error = new Error(res.error);
            throw error;
          }
        })
        .catch(err => {
          console.error(err);
          this.setState({ loading: false, redirect: true });
        });
    }
    render() {
      const { loading, redirect } = this.state;
      if (loading) {
        return null;
      }
      if (redirect) {
        return <Redirect to="/login" />;
      }
      return <ComponentToProtect {...this.props} />;
    }
  }
}
```

Рисунок 3.38 – Захищені маршрути React

Після цього створюємо модель сутності «Configuration», розробляємо додаткові методи для відтворення і збереження інформації, створюємо React-інтерфейс для відтворення інформації. Як результат отримуємо інтерфейс авторизації (рис. 3.17), інтерфейс трансляції відео потоку (рис. 3.18) та інтерфейс відображення вмісту сховища (рис. 3.19).

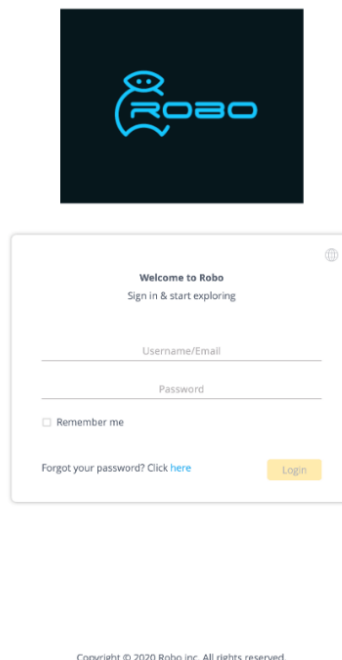


Рисунок 3.39- Інтерфейс авторизації

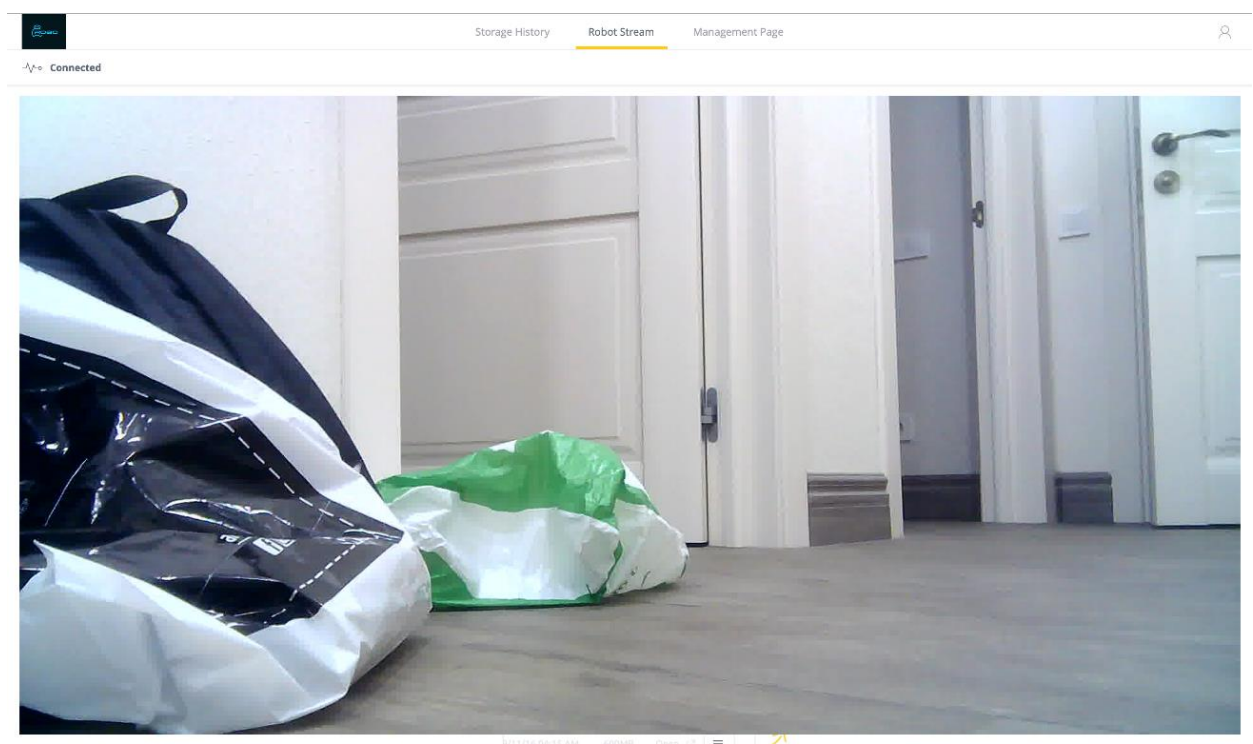


Рисунок 3.40 – Інтерфейс трансляції відео-потoku

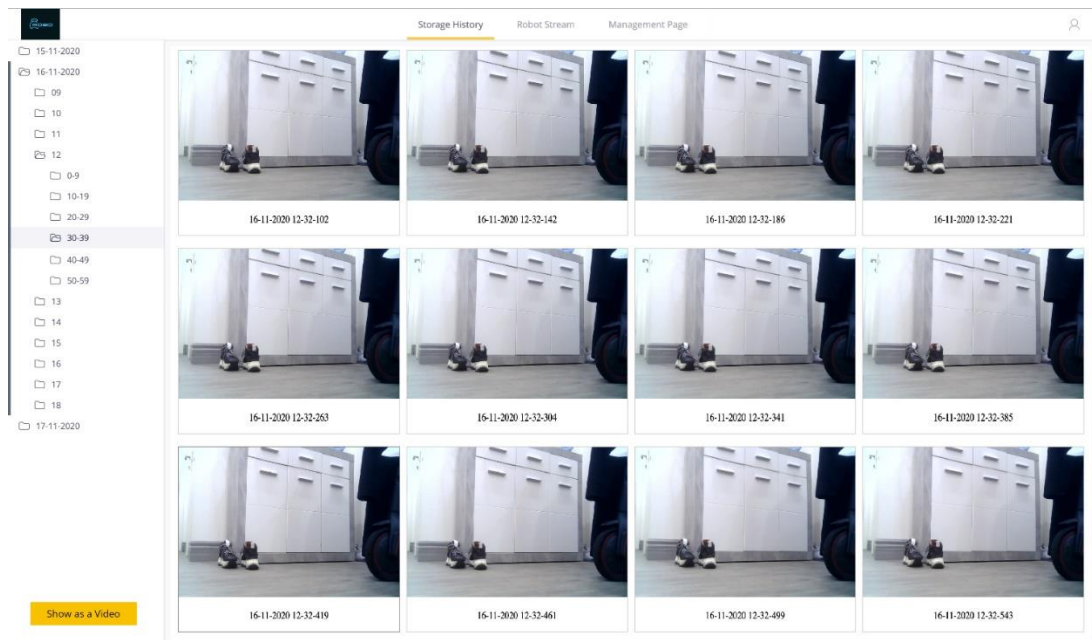


Рисунок 3.41 – Інтерфейс відображення вмісту сховища

Висновки до розділу

У цьому розділі було розглянуто основні методології проектування 3D-моделей прототипів, наведено переваги використання пакету САПР Fusion 360, спроектовано модель роботизованої рухомої охоронної платформи, а також розглянуто процес 3D-друку та процесу підготовки моделі до 3D-друку.

Було розглянуто роботу із інтерфейсом I2C у Raspberry Pi, проведено підключення усіх модулів і кодування алгоритмів роботи із ними.

Створено модуль авторизації для веб-платформи, розроблено модулі відображення прямої трансляції інформації з системи, а також записаних даних. Розроблено код для передачі і зберігання інформації на сервері.

4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

4.1. Опис ідеї проекту (товару, послуги, технології) (табл. 4.1.)

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Підвищити мобільність автоматизованих охоронних платформ за рахунок використання рухомої платформи	Контроль безпеки в індустріальній і домашній сфері	Економія людських і матеріальних ресурсів, підвищення безпеки виробництва і домашньої безпеки

Аналіз потенційних техніко-економічних переваг ідеї.

Таблиця 4.2. Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економічні характеристики ідеї
1.	Простий зрозумілий інтерфейс
2.	Функціональність
3.	Невеликий обсяг даних
4.	Гнучкість

Таблиця 4.2 Продовження

(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
Мій проект	Конкурент1	Конкурент2	Конкурент3			
Роботи зована рухома охоронна платформа	Стационарна охоронна станція з додатковими модулями	Система камер відеоспостереження із веб-платформою для контролю	Відсутній	Вразливість до різних умов середовища (поверхні для пересування, погодних умов)	Потрібне початкове мануальне конфігурування системи	Відсутність необхідності придбання додаткових модулів та їх встановлення
						Можливість тонкого конфігурування кожної точки контролю
						Зручна веб-платформа

						Велика швидкість передачі зображень та мала навантаженість файлового серверу
--	--	--	--	--	--	--

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність проекту

№ п/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1.	Створення рухомої, роботизованої охоронної платформи	3D-друк, Python, Microsoft Azure, NodeJS, React	Технологія наявна	Технологія доступна, проблем виготовлення не передбачається

Висновок – програмне і апаратне забезпечення доступне, передбачається відсутність проблем.

4.3 Аналіз ринкових можливостей запуску стартапу

Попередня характеристика потенційного ринку стартапу.

Проведення аналізу попиту.

Таблиця 4.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	2 (виробники відеокамер і систем контролю навколишньої середовища)
2	Загальний обсяг продаж	~200 000 ум.од
3	Динаміка ринку (якісна оцінка)	Стагнує
4	Наявність обмежень для входу (вказати характер обмежень)	Наявність стабільного Wi-Fi покриття у приміщенні
5	Специфічні вимоги до стандартизації та сертифікації	Отримання ліцензії від ЗОП
6	Середня норма рентабельності в галузі (або по ринку), %	120%

Висновок – умови ринку сприятливі.

Характеристика потенційних клієнтів.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Надання нового методу захисту	Підприємства та приватні особи	Різні вимоги з точки зору площі покриття і кількості точок контролю	Надійність, якість, стабільність.

Висновок – потенційні групи клієнтів визначені, особливості товару для груп зазначені, потенційність попиту достатня.

Аналіз ринкового середовища.

Таблиця 4.6. Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Вихід конкурентів на ринок.	Після виходу продукту на ринок можлива поява конкурентів	Нові інновації для приваблення нових клієнтів.
2.	Наявність перешкод	У приміщенні може з'явитися перешкода, яка обмежить рух платформи.	Створення алгоритму об'їзду перешкод
3.	Знищення БД	Раптова помилка БД та втрата усіх даних	Періодичне резервування даних на хмарній платформі.
4.	Кібератака	Перевантаження серверів додатку, що унеможливить роботу з продуктом	Завчасний аудит інформаційної безпеки, застосування резервних серверів

Таблиця 4.7. Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Новий ринок збуту	Розповсюджувати товар за межами країни.	Локалізація продукту на мову країни-покупця та початок рекламної компанії
2.	Співпраця з користувачем.	Додавання нового функціоналу щодо потреб конкретного споживача.	Відповідна зміна програмного продукту.

Висновок: існують загрози та можливості та відповідні шляхи реагування для запобігання та реалізації. Проект можна впроваджувати.

Аналіз пропозицій на ринку.

Загальні риси конкуренції на ринку

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика.	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції – чиста.	Один продавець пропонує товар, який не має близьких замінників	Пропонує нові функції, що є точнішими та більш оптимізованими
2. За рівнем конкурентної боротьби – міжнаціональний.	Ринок орієнтований на компанії зі всього світу	Локалізація продукту (переклад додатку на інші мови)
3. За галузевою ознакою – внутрішньогалузева	Внутрішньогалузева	Розвинути функціонал додатку для покриття інших можливостей.
4. Конкуренція за видами товарів: - товарно-видова.	Усі методи виконують дві функції – аналіз даних з сенсорів та оптимізацію зображень.	Можливість спрогнозувати більше виключних ситуацій за допомогою використання нейромереж, можливість покращити якість зображення та зменшення її ваги.
5. За характером конкурентних переваг - нецінова	Додаток використовується паралельно з системами відеоспостереження	Можливість інтеграції додатку до інших систем відеоспостереження
6. За інтенсивністю – не марочна	Товар немарочний бренд не відіграє велику роль, здійснюється вихід на ринок	Розвиток бренду для досягнення популярності по всьому світу

Висновок – конкуренція прийнятна.

Аналіз конкуренції в галузі за Майклом Потером.

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Виробники відеокамер спостереження	Бар'єром входження в ринок є	Постачальники мають вміти Підключати	Споживачі повинні вміти Користуватис	Відеокамери спостереження

	Виробники систем контролю навколишнього середовища	неготовність користувачів користуватися рухомими платформами для захисту	платформу до існуючої БД	я функцією технічної підтримки	Стаціонарні станції контролю
Висновки	Інтенсивність не є великою, оскільки конкуренти використовують кардинально різні підходи до рішення задачі	Можливість виходу існує: пристрій використовує принципово новий підхід Строки виходу на ринок: одразу після релізу готового продукту	Не диктують	Не диктують	Консерватизм Споживачів (Небажання користувачів розбиратися з роботою пристрою)

Висновок – можливість роботи з визначеним станом конкуренції прийнятна. Бар'єри входження на ринок – низькі.

Фактори конкурентоспроможності

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Функціональність	У конкурентів немає можливості переміщувати пристрої у просторі
2	Розмір даних	Розмір зображень та показань сенсорів є одним з основних факторів конкурентоспроможності, для стартапу він є особливим через те, що він є джерелом даних.
3	Стабільність	Здатність пристрою працювати без збоїв протягом довгого періоду часу

Висновок – продукт має переваги та кращі характеристики певних параметрів.

Сильні та слабкі сторони

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін «роботизованої рухомої охоронної платформи»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з РРОП						
			-3	-2	-1	0	1	2	3
1	Функціональність	10		*					
2	Розмір даних	17	*						
3	Стабільність	14				*			

Висновок – РРОП має найвищу кількість балів

SWOT аналіз, виділення сильних та слабких сторін

Таблиця 4.12. SWOT – аналіз стартап-проекту

<p>Сильні сторони:</p> <ul style="list-style-type: none"> – Переміщення за допомогою рухомою платформи – Передбачення критичних ситуацій (нейромережі) – Висока швидкість передавання даних – Висока якість зображень 	<p>Слабкі сторони:</p> <ul style="list-style-type: none"> – Вихід конкурентів на ринок – Неготовність користувачів користуватися рухомими платформами для захисту
<p>Можливості:</p> <ul style="list-style-type: none"> - Інтеграція додатку в старі системи автоматизації та спостереження 	<p>Загрози:</p> <ul style="list-style-type: none"> - Складний процес інтеграції з продуктом

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Інтеграція додатку в старі системи автоматизації та спостереження.	Висока ймовірність (75%)	Середні строки реалізації (2 місяці)
2	Ефективна реклама продукту, що зацікавить клієнтів	Середня ймовірність (50%)	Середні строки реалізації (2 місяці)

4.4 Розробка ринкової стратегії проекту

Таблиця 4.14 Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Підприємства із сучасним обладнанням і високим рівнем	Середня Підприємства з високим рівнем автоматизації	Високий Заміна стаціонарних станцій на	Конкуренція серед стаціонарних систем	Середня Це потребує особистих продажів і

	автоматизації виробництва	використовують стаціонарні сенсорні і відеосистеми.	рухомі здатний заменшити витрати на закупівлю і збільшити кількість контрольованих зон.	контролю. Інтенсивність мала	відвідувань профільних заходів
2	Фанати IoT і Smart Home	Середня Фанати IOT з великим інтересом ставляться до роботів/рухомих платформ	Низький Зі збільшенням кількості електроніки у будинку – пожежна охорона є дуже актуальною, однак вартість продукту завелика	Конкуренція серед «розумних» пилососів, стаціонарних систем стеження. Інтенсивність Середня.	Легка У цьому сегменті дуже добре працює масмаркетинг
Які цільові групи обрано: підприємства					

Таблиця 4.15 Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Ринок: підприємства та забудовники	Пропозиція нового типу захисту	Оптимальне співвідношення роботи алгоритмів і функціональності	Стратегія диференціації

Таблиця 4.16 Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Так	Ні. Мій програмний продукт може працювати поряд з	Ні	Стратегія Лідера

		системами відеоспостереження		
--	--	---------------------------------	--	--

Таблиця 4.17. Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Функціональність, точність та швидкість роботи	Стратегія диференціації	Мінімізація розміру фотографій без особливої втрати якості	Зручність використання Висока швидкість передачі даних Підтримка товару (support)

За базову стратегію розвитку обрано стратегію диференціації.

4.5 Розроблення маркетингової програми стартап-проекту

Спершу формується маркетингова концепція товару для споживача (табл. 4.18).

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Новий тип захисту з новими можливостями	Велика швидкодія, економія ресурсів	Рухома платформа дозволяє збільшити кількість точок контролю та зменшити кількість витрат на стаціонарні системи

Наступною відбувається розробка трьохрівневої маркетингової моделі товару: уточнення ідеї продукту, його фізичні складові (табл. 4.19).

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за заду- мом	Охоронна система повинна бути впровадженою зі швидкістю інтернету 1 Гб/с, мати 1 Тб пам'яті БД, передавати 10 фотографій в одиницю часу, рухатись зі швидкістю 1 м/с, мати роздільну здатність 12 Мп, мати кут огляду 360°, дальність ІЧ-підсвічування 20м.		
II. Товар у	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор

реаль- виконанні	ноту	1. Керованість	М	Тх
		2. Об'єм інформації, що зберігається	М	Тх
		3. Швидкість оптимізації	М	Тх
		Якість: Гарантована висока якість		
		Пакування: доступ до репозиторію та рухома платформа		
		Марка: PPOC		
III. Товар із підкріпленням	Існує технічна підтримка товару, яку можна викликати для налагодження роботи, виправки несправностей.			
За рахунок чого потенційний товар буде захищено від копіювання: за рахунок новітніх ідей				

Наступним етапом є визначення рівня цін на потенційний товар (табл. 4.20).

Таблиця 4.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на Товари-замінники	Рівень цін на Товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	Відеокамери (мін. Ціна 3500 грн.)	Товари-аналоги відсутні на ринку, тож рівень цін неможливо встановити.	Вільно високий Забудовники та підприємства здатні виділяти великі бюджети в залежності від доцільності використання технології.	Продаж продукту: 1000\$ Технічна підтримка: 250\$ за виклик

Наступним етапом є формування системи збуту (табл. 4.21).

Таблиця 4.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	Клієнти (підприємства та забудовники), які прагнуть підвищити рівень захисту	Функція Billing Функція refund	Глибина 1: для окремих користувачів	Сервісна модель розповсюдження без Альтернативних каналів збуту

Останнім етапом є розробка концепції маркетингових комунікацій (табл. 4.22).

Таблиця 4.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Нерегулярні продажі	Гаряча лінія, персональний кабінет	Функціональність, інноваційність	Мотивація Клієнта використовувати даний продукт	With our innovative smart security system, you'll always be aware of what is going on in your house

Висновки до розділу

У цьому розділі визначені ідеї стартапу, його сильні та слабкі сторони, здійснений технологічний аудит проекту, здійснене дослідження ринку, цільових груп клієнтів, їх характеристика, очікування та бажання.

Проведений SWOT-аналіз, аналіз конкуренції, визначені стійкі конкурентоздатні властивості та переваги, розроблена ринкова стратегія та маркетингова програма.

Дослідження встановило, що продукт має унікальні характеристики, існують сегменти споживачів на ринку, які можуть бути зацікавлені в придбанні товару. Використовуючи стратегію диференціації як стратегію розвитку та стратегію лідера як базову стратегію конкурентної поведінки можна виходити на ринок.

ВИСНОВКИ

В першому розділі цієї магістерської дисертації було визначено об'єкт і предмет дослідження, проведено аналіз існуючих систем охорони домашнього та індустріального масштабу, виявлено їх основні недоліки і переваги. Проаналізовано актуальний перелік модулів і технологій, які доцільно використовувати в цьому проекті. Також було проведено огляд технологій проектування і прототипування, які значно полегшують процес розробки фінального продукту. На основі аналізу існуючих систем було сформовано основні причини актуальності дослідження і були поставлені задачі, які необхідно буде вирішити.

Далі було проведено аналіз принципу роботи систем пересування роботів, проведено аналогію із біологічними організмами, розглянуто основні види доступних платформ пересування роботів, визначено найкращий для цього проекту. Також було розглянуто існуючі на ринку одноплатні комп'ютери, було обрано найкращий варіант по співвідношенню вартість/виконувані функції та по можливості досягнути усі поставлені задачі. У останньому пункті другого розділу було розглянуто переваги 3D-друку як потужного інструмента прототипування як для ентузіаста, так і в промислових масштабах. Визначено основні види 3D-друку, їх переваги і недоліки.

У практичній частині дисертації розглянуто процес проектування 3D-моделі платформи, сформовано архітектуру апаратної і програмної частини системи, продемонстровано процес підключення і програмування усіх модулів та сенсорів, розроблено програмний код для передачі даних на сервер, а також розглянуто програмний код Back-End і Front-End частини системи. Результатом виконаної роботи є система, яка виконує поставлені перед нею задачі. Апаратна платформа є стійкою та сбалансованою, система збирає інформацію з сенсорів і успішно відправляє їх на сервер.

У останньому розділі було проведено маркетинговий аналіз стартап-проекту. Визначені ідеї стартапу, його сильні та слабкі сторони, здійснений

технологічний аудит проекту, здійснене дослідження ринку, цільових груп клієнтів, їх характеристика, очікування та бажання

ПЕРЕЛІК ПОСИЛАНЬ

1. What is Arduino? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.arduino.cc/en/Guide/Introduction>
2. Design the robot as security system in the home / Gašparík Marek*, Šolek Peter, 2014
3. Configuring I2C [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.adafruit.com/adafruit-raspberry-pi-lesson-4-gpio-setup/configuring-i2c>
4. Dagu Wild Thumper [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pololu.com/product/1561>
5. RPi.bme280 [Електронний ресурс] – Режим доступу до ресурсу: <https://pypi.org/project/RPi.bme280/>
6. Adafruit DC and Stepper Motor HAT for Raspberry Pi [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.adafruit.com/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi>
7. Express/Node introduction [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction

ДОДАТКИ

ДОДАТОК А

Результат перевірки на співпадіння



User name:
Лісовиченко Олег Іванович

Check ID:
1005364708

Check date:
04.12.2020 14:29:20 EET

Check type:
Doc vs Internet + Library

Report date:
04.12.2020 14:30:47 EET

User ID:
76913

File name: **Машковцев Олександр ІК-91МП**

Page count: **41** Word count: **9259** Character count: **66412** File size: **82.91 KB** File ID: **1005657279**

0.91% Matches

Highest match: **0.29%** with Library source (File ID: **1005657274**)

0.39% Internet sources

9

Page 43

0.91% Library sources

120

Page 43

0% Quotes

Exclusion of quotes is off

Exclusion of references is off

0% Exclusions

No exclusions

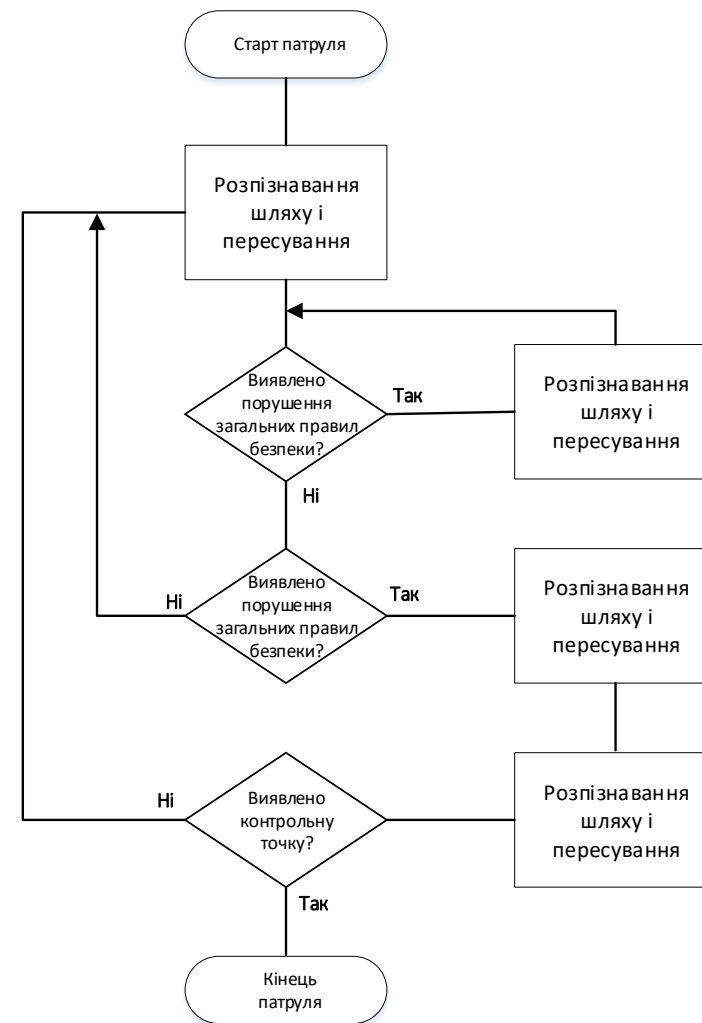
ДОДАТОК Б

Результат перевірки на співпадіння

ДОДАТОК В

Плакати

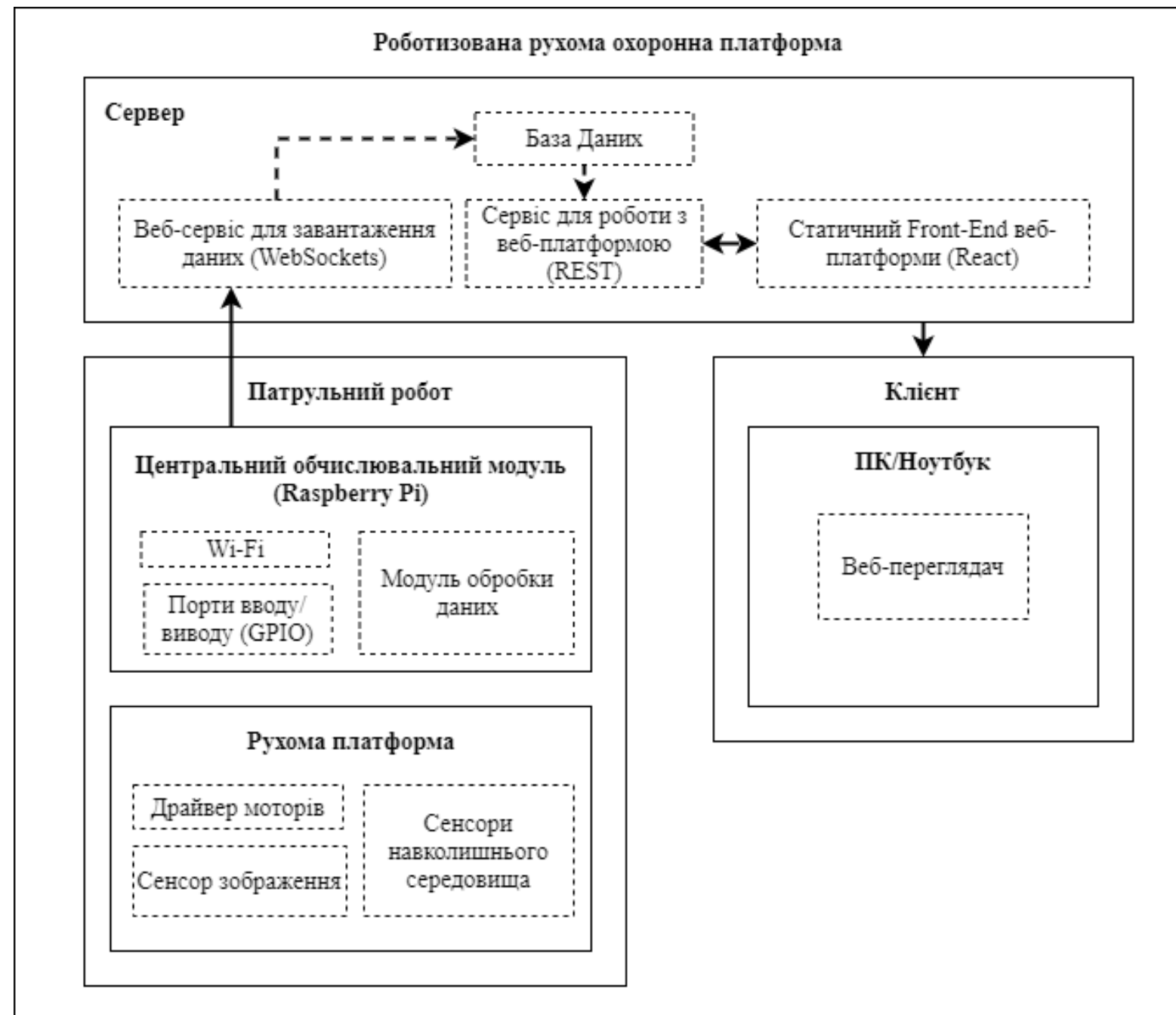
Блок-схема алгоритму патрулювання



Демонстраційний плакат №1 «Блок-схема алгоритму патрулювання»
до дипломної роботи на тему
«Роботизована рухома охоронна платформа»

Виконав: студент гр. ІК-91мп Машковцев О.С.
Керівник: к. т. н., доцент Крилов В.П.

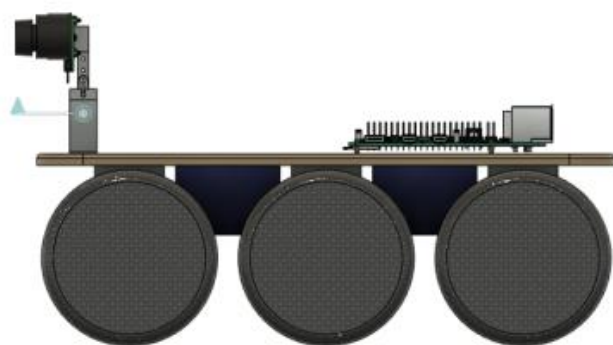
Архітектура системи



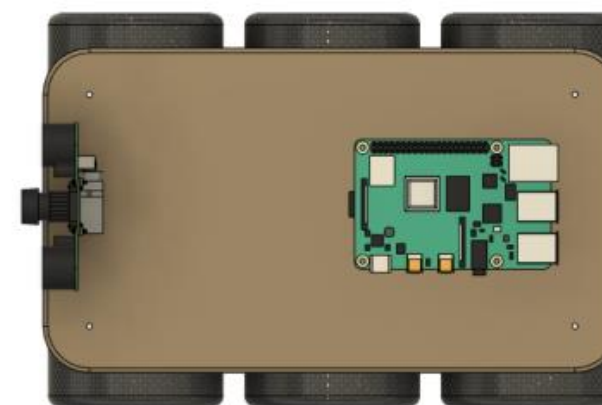
Демонстраційний плакат №2 «Архітектура системи»
до дипломної роботи на тему
«Роботизована рухома охоронна платформа»

Виконав: студент гр. ІК-91мп Машковцев О.С.
Керівник: к. т. н., доцент Крилов В.П.

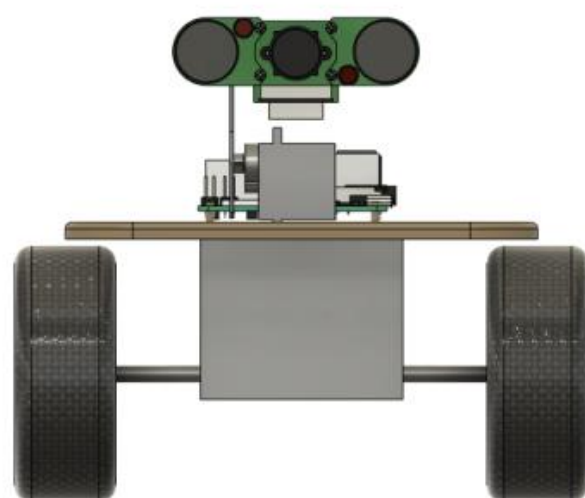
3D-модель платформи



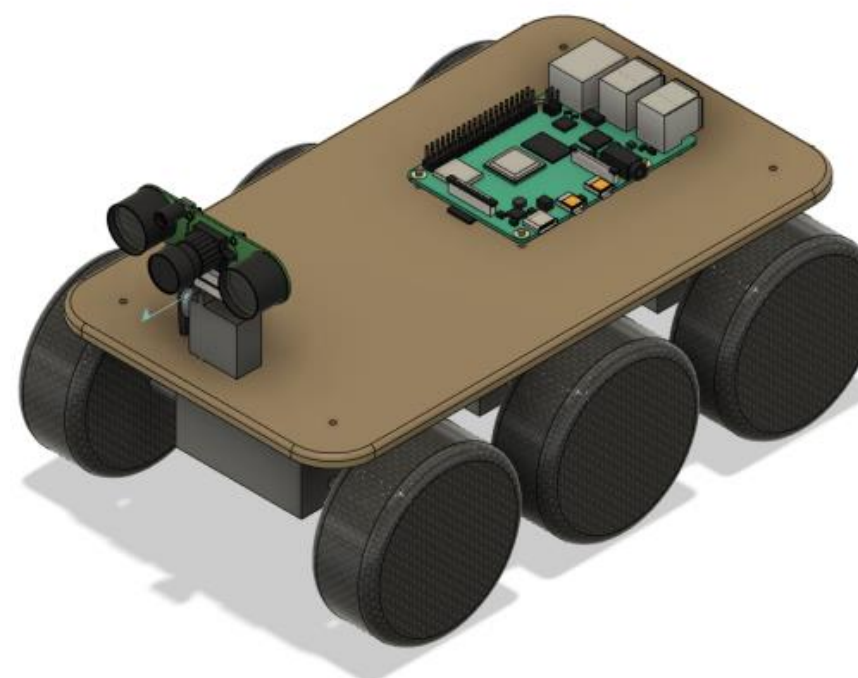
3D-модель платформи. Проекція зліва



3D-модель платформи. Проекція зверху



3D-модель платформи. Проекція спереду



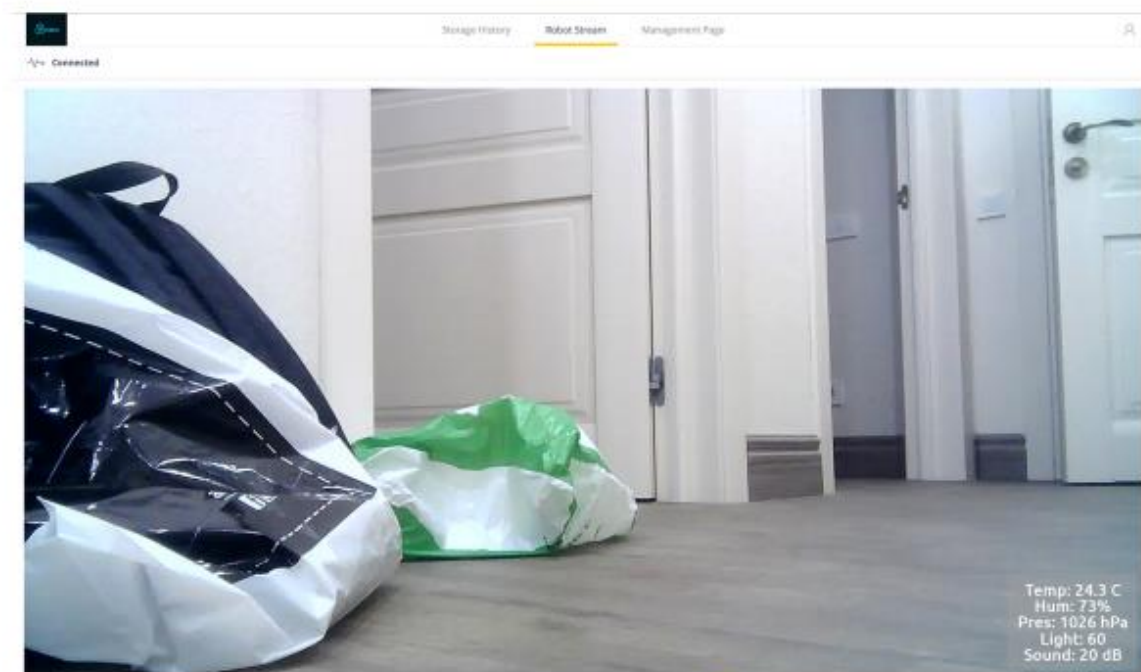
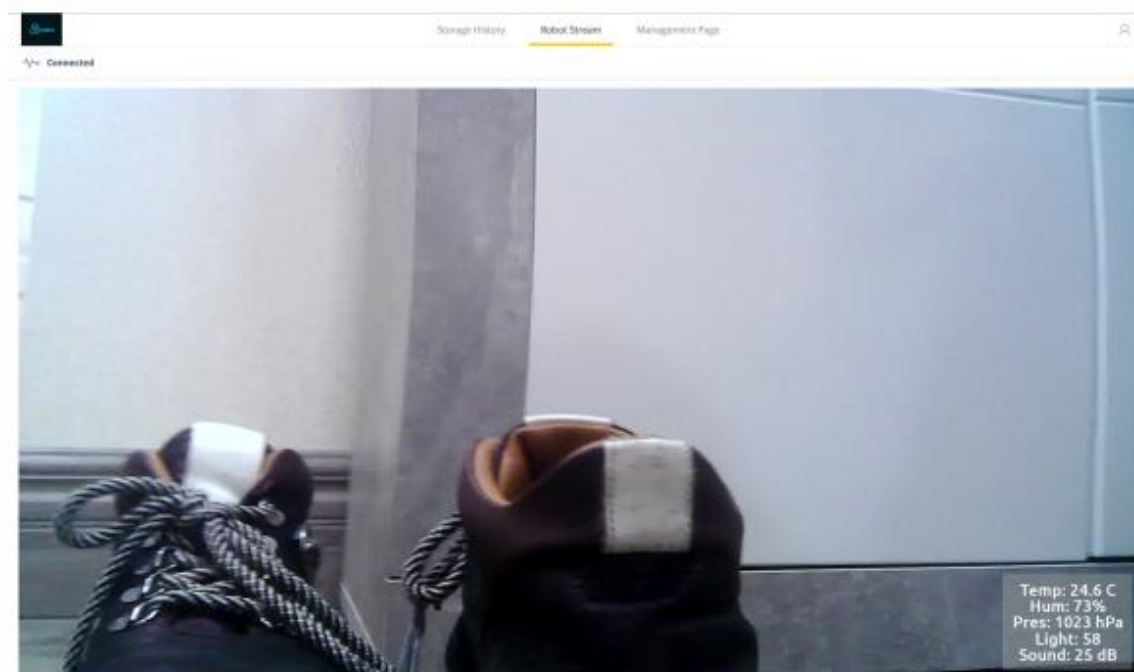
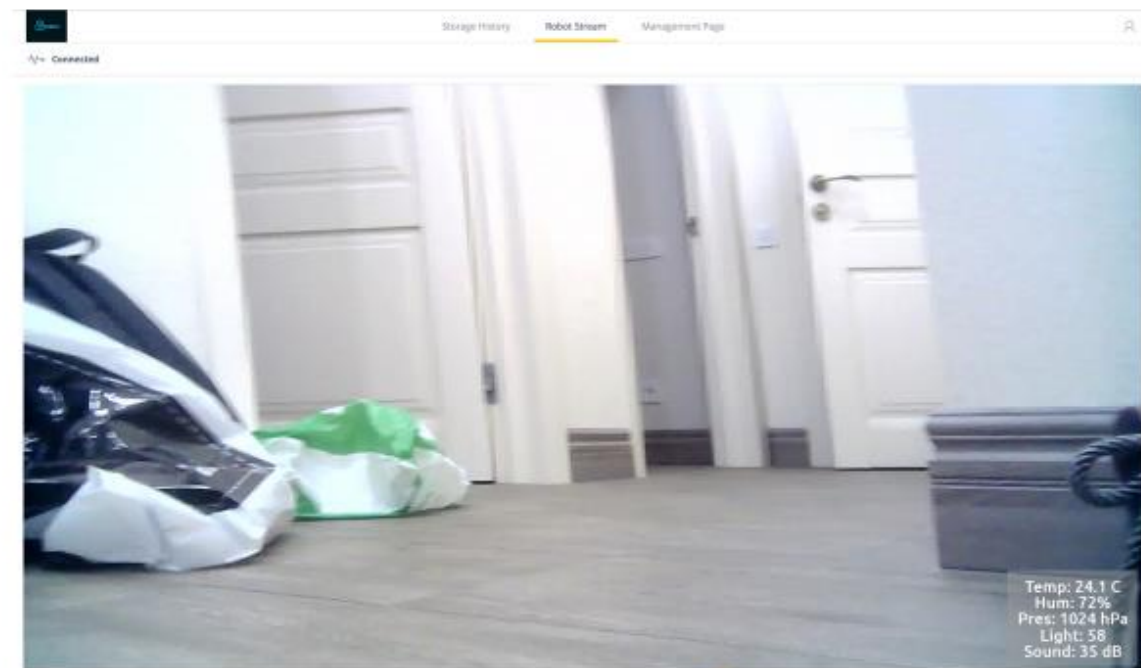
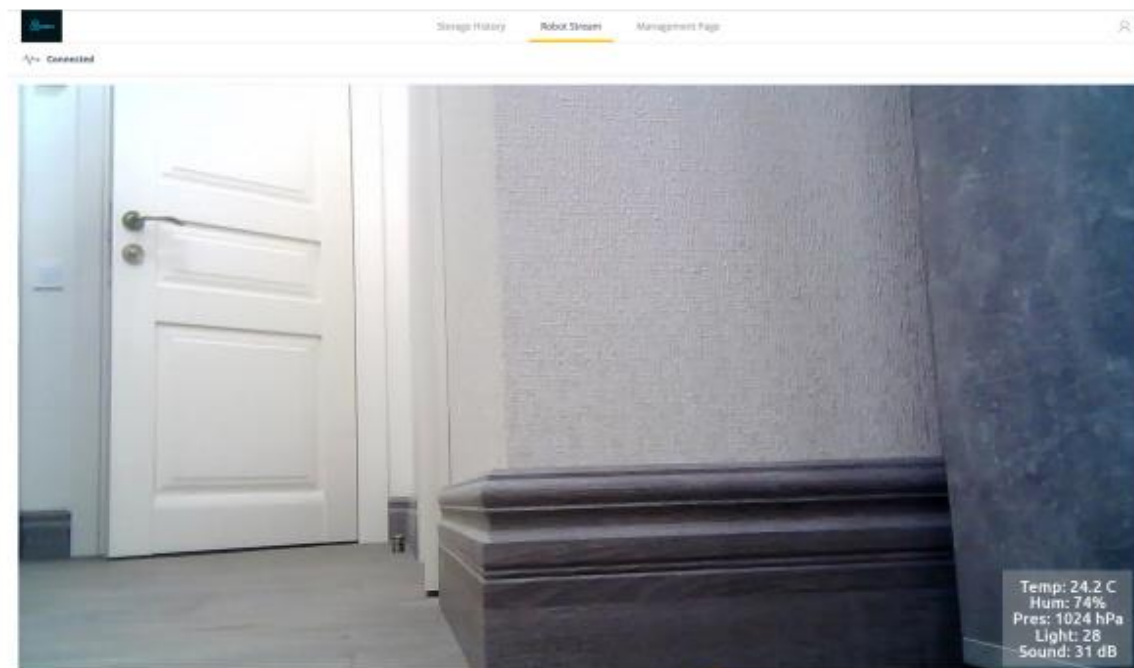
3D-модель платформи

Демонстраційний плакат №3 «3D-модель платформи»
до дипломної роботи на тему
«Роботизована рухома охоронна платформа»

Виконав: студент гр. ІК-91мп Машковцев О.С.
Керівник: к. т. н., доцент Крилов В.П.

Виконав: студент гр. ІК-91мп Машковцев О.С.
Керівник: к. т. н., доцент Крилов В.П.

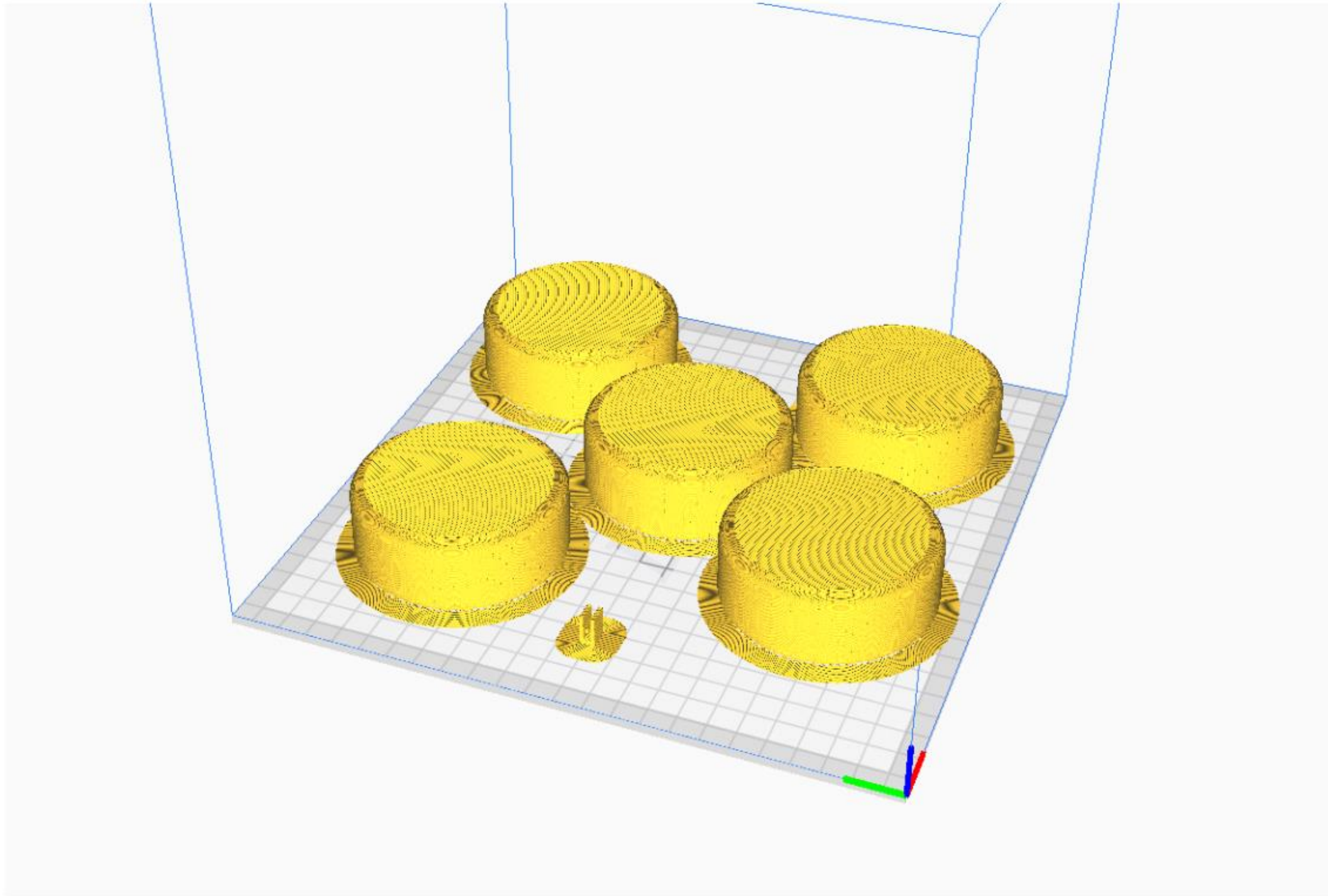
Інтерфейс відеопотоку



Демонстраційний плакат №5 «Інтерфейс відеопотоку»
до дипломної роботи на тему
«Роботизована рухома охоронна платформа»

Виконав: студент гр. ІК-91мп Машковцев О.С.
Керівник: к. т. н., доцент Крилов В.П.

Візуалізація GCODE Для 3D-принтеру



Демонстраційний плакат №6 «Візуалізація GCODE Для 3D-принтеру»
до дипломної роботи на тему
«Роботизована рухома охоронна платформа»

Виконав: студент гр. ІК-91мп Машковцев О.С.
Керівник: к. т. н., доцент Крилов В.П.